COMMODORE USERS

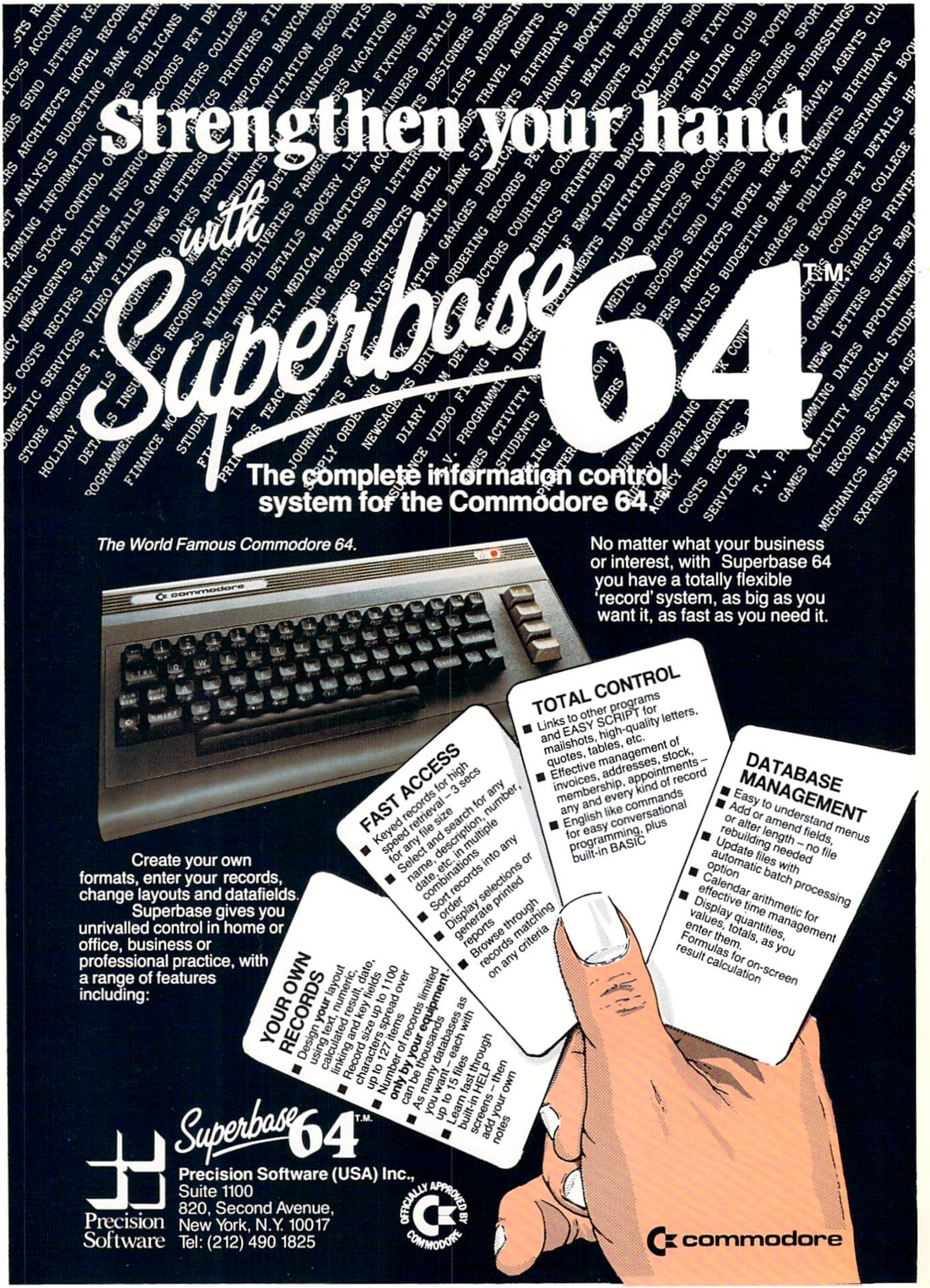
Ahoy!

48355
K

$2.50/CAN.$2.75 JULY 1984



07
0 71896 48355


Strengthen your hand
with Superbase 64 T.M.
The complete information control system for the Commodore 64.
The World Famous Commodore 64.
No matter what your business or interest, with Superbase 64 you have a totally flexible 'record' system, as big as you want it, as fast as you need it.
Create your own formats, enter your records, change layouts and datafields. Superbase gives you unrivalled control in home or office, business or professional practice, with a range of features including:
YOUR OWN RECORDS
Design your layout using text, numeric, calculated result, date, linking and key fields
Record size up to 1100 characters spread over up to 127 items
Number of records limited only by your equipment. Can be thousands
As many databases as you want – each with up to 15 files
Learn fast through built-in HELP screens – then add your own notes
FAST ACCESS
Keyed records for high speed retrieval – 3 secs for any file size
Select and search for any name, description, number, date, etc. in multiple combinations
Sort records into any order
Display selections or generate printed reports
Browse through records matching on any criteria
TOTAL CONTROL
Links to other programs and EASY SCRIPT for mailshots, high-quality letters, quotes, tables, etc.
Effective management of invoices, addresses, stock, membership, appointments – any and every kind of record
English like commands for easy conversational programming, plus built-in BASIC
DATABASE MANAGEMENT
Easy to understand menus
Add or amend fields, or alter length – no file rebuilding needed
Update files with automatic batch processing option
Calendar arithmetic for effective time management
Display quantities, values, totals, as you enter them.
Formulas for on-screen result calculation
Superbase 64 T.M.
Precision Software
Precision Software (USA) Inc.,
Suite 1100
820, Second Avenue,
New York, N.Y. 10017
Tel: (212) 490 1825
OFFICIALLY APPROVED BY COMMODORE
commodore

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Cover illustration by James Regan
(inside illustration by Kevin Walter)

Publisher
Michael Schneider

Editor
David Allikas

Managing Editor
Robert J. Sodaro

Senior Editor
Tim Moriarty

Consulting Editors
Ben Bova
Morton Kevelson
Dale Rupert

Art Director
Joan Mazzeo-Weinshank

Associate Art Director
Raoul Tenazas

Assistant Art Director
Jo Ann Case

Art Production
Christopher Carter
Pauline Giordani
Eve Griffin

Technical Advisor
Edward D. Laufer

Circulation Director
W. Charles Squires

Advertising Director
Lynne Dominick

Director of Promotion
Joyce K. Fuchs

Controller
Dan Tunick

Managing Director
Richard Stevens

Advertising Representative
JE Publishers Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213) 467-2266
Boston (617) 437-7628
Dallas (214) 660-2253
New York (212) 724-7767
Chicago (312) 445-2489
Denver (303) 595-4331
San Francisco (415) 864-3252

AHOY! No. 7, July 1984. Published monthly by Ion International Inc., 45 W. 34th St., Suite 407, New York, NY, 10001. Subscription rate: 12 issues for $19.95, 24 issues for $37.95 (Canada $26.95 and $49.95 respectively). Application to mail at 2nd Class postage rates is pending at New York, NY and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street—Room 407, New York, NY 10001. All editorial and subscription inquiries and software and hardware to be received should be sent to 45 W. 34th St., Suite 407, New York, NY 10001.

VIEW FROM THE BRIDGE

Our *View from the Bridge* space has been severely shortened this time around by an embarrassingly long list of program corrections. We'll therefore skip the chatty opening and get right to this month's *Ahoy!*

It's seldom that we devote feature space to a single product. But we think you'll approve of our departure from usual procedure after reading *The MSD SD-2: An Indepth Report.* Morton Kevelson takes the new disk drive apart, and puts the pieces together for you. (Turn to page 42.)

The one software package that everyone gets around to owning sooner or later is a database. But not everyone winds up owning the database that's right for him. In *Database Buyer's Guide,* Larry Bihlmeyer surveys some of the choices available for the VIC and 64. (Turn to page 25.)

We haven't run a users group feature since our February issue. Not because of lack of interest, but because we were looking for the right author to begin our new semi-regular column devoted to the subject. We found Joycelyn Woods, membership secretary of New York City's leading Commodore users group. She'll tell you all about *Starting or Joining a Users Group.* (Turn to page 21.)

Michael Kleinert and David Barron have returned to teach you how to manipulate your cursor in undreamed of ways. (Turn to page 37.)

Richard Herring continues his six-part *Educational Software* series. *Part V* contains some of the most useful information yet provided for parents of aspiring scholars. (Turn to page 63.)

We know you'll find the two utility programs featured this issue particularly useful. Bob Spirko's *Checklist* will help even the average C-64 owner to put order into his life. (Turn to page 19.) And *Renumbering Utility* by Anthony Wood is as useful as it is short. (Turn to page 29.)

As for games, we've got them, too. (We sure wouldn't bring them up if we *didn't* have them!) *Brisk* by Joseph Fusco will let you match card-sharking prowess with your Commodore 64. (Turn to page 41.) *What's My Job?* by B.W. Behling (author of *Easy Access Address Book* and *I Got Biorhythm)* will, as the saying goes, amaze your friends with its ability to guess their occupations. (Turn to page 11.) And Andy Van Duyne's *Math Defender* will let kids and adults alike bone up on their arithmetic while experiencing real arcade game excitement. (Turn to page 47.)

You'll find all our regular features inside as well. That means *Creating Your Own Games on the VIC and 64* wherein Orson Scott Card talks about designing the kind of compelling screen displays you need to hold a player's interest (turn to page 13); *The Rupert Report,* in which Dale Rupert launches into an exhaustive discussion of sequential files (turn to page 59); and *Commodares,* in which Dale unwinds from all that programming by teasing the brains of *Ahoy!'s* most daring readers.

We'll permit ourselves just one bit of chat. How do you like our cover? We hope you do, because you're going to be seeing a lot of it. All future issues of *Ahoy!* will feature that design, with different specifics inserted in the screens.

Thanks for coming onboard once again!

—David Allikas

PROGRAM CORRECTIONS!

If you missed last issue, be advised that new versions of our *C-64* and *VIC 20 Bug Repellents* were introduced. If you'd previously typed in the versions that appeared in our April or May issues, you must update them to conform to the versions appearing last month and this month.

Other program corrections are as follows:

Regarding *Sound Concept,* the program lines provided on page 98 that must be incorporated into all outside BASIC programs that make use of *Sound Concept* should have concluded as follows:

```
62000 POKE 1,INT (AD/256):POKE 0,
 AD - 256 * PEEK (1):POKE 828,1:P
OKE 782,3:RETURN
```

The relevant lines of Test 1 of *Memory Management* should be corrected to read as follows:

```
20 IFPEEK(251)=1GOTO90
50 GOSUB200:GOSUB300
70 POKE44,192:REM FOR VIC POKE 44
,4
200 PRINTPEEK(43);PEEK(44);PEEK(4
5);PEEK(46)
220 PRINTPEEK(51);PEEK(52);PEEK(5
5);PEEK(56)
230 RETURN
```

Biorhythm contains a slight glitch which can be eliminated by correcting the relevant lines to read as follows (also, delete lines 155 and 255):

```
45 IFLEN(BI$)<>10ORLEN(DA$)<>10TH
ENGOTO275
170 GOTO162
250 PRINT"{CD}{CD}{CD} HIT ANY KE
Y TO CONTINUE."
265 DM$=STR$(CM%):DD$=STR$(CD%):D
Y$=STR$(CY%)
270 GOTO49
```

SCUTTLEBUTT

LETTER-QUALITY PRINTERS FROM CARDCO AND AMDEK • C-64 MEMORY EXPANDER • COMPUTER FURNITURE • SPRITE PROGRAM • ACCOUNTING SYSTEM • NEW GAME RELEASES FROM ACTIVISION, MUSE, INFOCOM AND CBS • KNOWLEDGE ENGINEERING • BILLBOARD'S HOTTEST SELLERS

ALL THE PRINTER NEWS THAT'S FIT TO PRINT

Cardco, long known for producing interfaces that connect the C-64 and VIC 20 to non-Commodore printers, now seem intent on cutting into their own sales with a line of letter-quality printers that connect to Commodore computers without an interface.

The LQ/1, already seen in our ad pages, features standard friction-feed and optional tractor-feed. Either serial or lineprint modes can be selected by means of a simple DIP switch operation. Printing speed is 14 characters per second; pitch can be 10, 12, or 15. Price is $649.95.

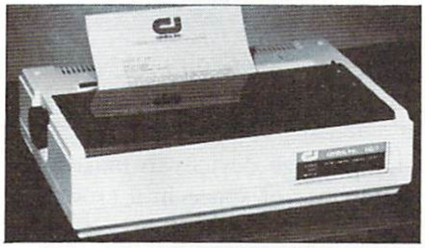
The LQ/3, priced at $449.95, includes many of the features of the LQ/1. As with the LQ/1, its ESC sequence allows minimum increments of $\frac{1}{20}$ inch for carriage movement and $\frac{1}{48}$ inch for paper feed, making bold print, superscript, subscript, and underlining functions possible when used with an advanced word processor. Speed is 13 characters per second; pitch is 10.

The more compact LQ/2 ($349.95) utilizes a rotating character drum instead of a daisy wheel. It is a true portable, weighing only 6.6 pounds and featuring an optional internal battery. Printing speed is 13 characters per second.

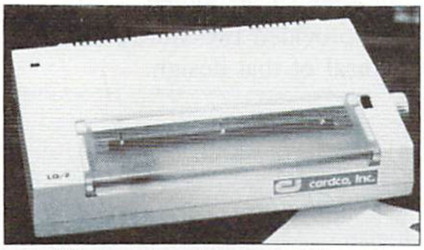
All three printers feature bidirectional printing with logic-seeking and a character set of 96.

Cardco, Inc., 313 Mathewson, Wichita, KS 67214 (phone: 316-267-6525).

Amdek Corporation has also released three new letter quality printers, including a portable (the 5025) that prints at 25 characters per second and two desktops: the 5040 (40 characters per second)



LQ/1: serial and lineprint modes. READER SERVICE NO. 17



LQ/2: optional internal battery. READER SERVICE NO. 18



LQ/3: has many features of LQ/1. READER SERVICE NO. 19

and the 5055 (55 characters per second). As with automobiles, the price of increased speed is a high one: the printers retail for $799, $1675, and $1990 respectively. All include built-in Centronics Parallel and RS-232 interfaces.

Amdek Corporation, Printer Products, 2201 Lively Blvd., Elk Grove Village, IL 60007 (phone: 312-364-1180).

INCREASED C-64 MEMORY

Sure, 64K is the accepted microcomputer industry standard—but since when have industry standards meant anything to Commodore 64 owners? Leader Electronic Technology Company (LETCO), who gave us the first 64K RAM module for the VIC (the 64KV) have developed a 64K memory



Double your C-64's memory. READER SERVICE NO. 20

expander for the 64. The 64KC RAM memory expander plugs directly into the 64's expansion port, providing the user with eight separate 8K blocks of paged memory that can each be selected by a single POKE instruction.

Owners of the 64KV need only the 64KVA adaptor to use their VIC expander on the 64. The 64KVA sells for $29.95, the 64KV for $109.95, and the 64KC for $139.95. LETCO is also developing an expander that will allow up to 256K of memory.

LETCO, 7310 Wells Road, Plain City, OH 43064 (phone: 614-873-4410).

COMMODORE'S GREATEST HITS

Once again, we provide Billboard Magazine's ranking of top-selling computer software in the categories of Entertainment, Education, and Home Management, for the week ended 4/28/84. Only the Commodore compatibles in the Top 10 or 20 are listed. Copyright 1984 by Billboard Publications, Inc., compiled by the Billboard Research Department, and reprinted with permission.

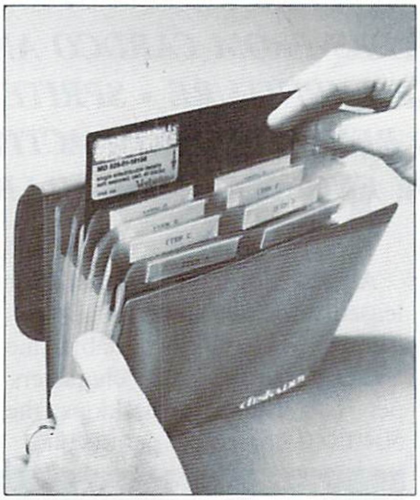
Entertainment: *Flight Simulator II,* Sublogic (#1); *Zork I,* Infocom (#3); *Beach-Head,* Access (#4); *Archon,* Electronic Arts (#8); *Pinball Construction Set,* Electronic Arts (#9); *Mystery Master: Murder by the Dozen,* CBS Software (#11); *Deadline,* Infocom (#12); *Zaxxon,* Synapse (#13); *Blue Max,* Synapse (#16); *Choplifter,* Broderbund (#17); *Hard Hat Mack,* Electronic Arts (#18); *Sorcerer,* Infocom (#20).

Education: *Mastertype,* Scarborough (#1); *Music Construction Set,* Electronic Arts (#2); *KinderComp,* Spinnaker (#3); *Computer Study Program for the SAT,* Barron's Educational Series (#4); *Facemaker,* Spinnaker (#5); *Spellicopter,* Designware (#6); *M.U.L.E.*, Electronic Arts (#7); *Type Attack,* Sirius (#9); *Story Machine,* Spinnaker (#10).

Home Management: *The Home Accountant,* Continental (#1); *The Tax Advantage,* Continental (#4); *Multiplan,* HesWare (#6); *Homeword,* Sierra On-Line (#8); *Simons' BASIC,* Commodore (#9); *Paperclip,* Batteries Included (#10).

FURNITURE AND STORAGE DEVICES

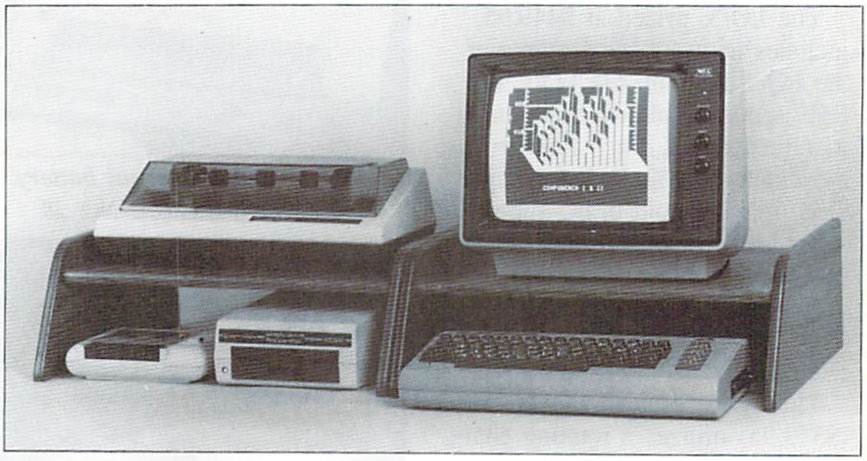
In the computer era, the simple flat desk has gone the way of parchment and quill pens. There is now an array of new furniture and storage devices to make maximum use of space in a crowded world.



Diskaddy holds up to 22 diskettes. READER SERVICE NO. 21

The fundamental design for tabletop computer shelves seems to be established; Sound Expansion has provided two sturdy models typical of that design.

Sound Expansions' Compubench I has one shelf and measures 7″ in height, 18″ depth, and 19¾″ in width. It retails for $49.95. The Compubench II, with two shelves, is 12″ in height, 18″ in depth, and 19¾″ in width and sells for $69.95. Both Compubenches are made of solid oak. They are deeper than many computer tables of this design, mainly for the purposes of consolidating wires, power packs, and expansion boards.



Both the one-shelf (pictured) and two-shelf Compubenches are solid oak. READER SERVICE NO. 22

Sound Expansions, PO Box 53, House Springs, MO 63051.

HollisCraft's DeskMate features a rotating turntable for the monitor. Also, there is a slot above the second shelf to feed fanfold paper. The DeskMate is made of pine, weighs 15 lbs., and measures 36″ in width, 13″ in height and 12″ in depth. It retails for $44.95.

HollisCraft, PO Box 465, Oakdale, MA 01539.

Suncom is also getting into the act. Their System Station's basic unit consists of a desk at typing table height on four casters. This basic unit sells for $79.00. Optional add-ons include a detachable monitor/disk drive stand and a printer/paper, diskette, book storage unit.

Suncom Inc., 650 Anthony Trail, Unit E, Northbrook, IL 60062.

Although Gusdorf has no new products of which we're aware, no computer furniture piece would be complete without a mention of them. They've been at it longer than most and have a wide array of products.



HollisCraft's DeskMate features a rotating turntable for the monitor.
READER SERVICE NO. 23

Gusdorf, 6900 Manchester Ave., St. Louis, MO 63143.

Close Enterprises has introduced two models of a disk storage unit. The Disk-O-Tel Model DT (desk top) contains dust-proof room for 108 disks and is mounted on a ball-bearing turntable. The high-rise Disk-O-Tel is the same unit except that it is mounted on a swivel arm. When attached by clamp to the desk edge, the unit sits eight inches above the desk surface. The high-rise retails for $59.95, the desk top for $49.95.

Close Enterprises, PO Box 13903, Dept. 40, Arlington, TX 76013.

Finally, for the on-the-go (or on-the-run) programmer, MB International has provided the Diskaddy, which holds up to 22 5¼" diskettes. Prices for the Diskaddy range from $7.99 to $21.99.

MB International, Inc., 701 B Street, Suite 1300, San Diego, CA 92101.

WATCH THAT NEST EGG

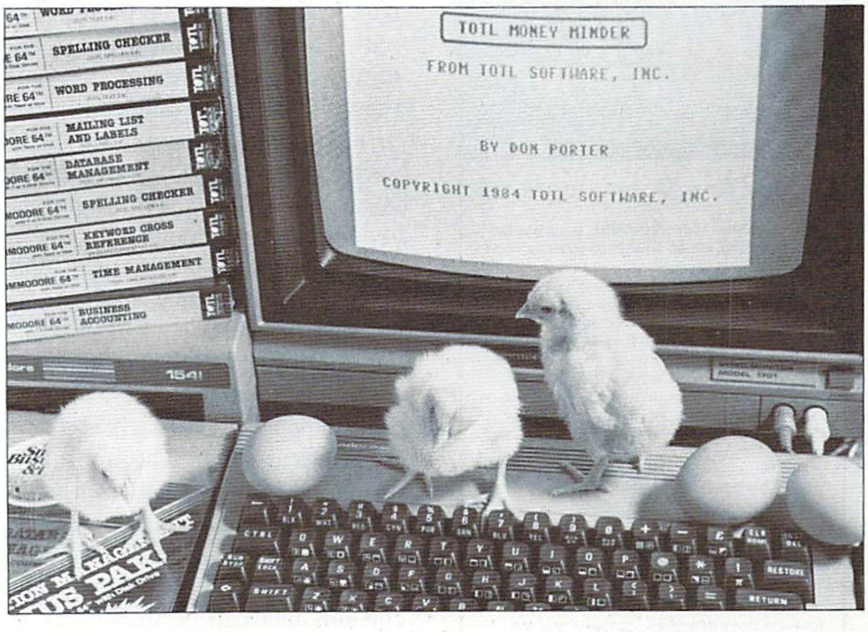
TOTL. Moneyminder is the latest C-64 release from the chicken outfit that brought you the variety of home and small business software pictured. This, a complete home accounting system, allows up to 110 user-defined accounts, with double entry transactions that can be spread across multiple expense accounts. It will record up to 8400 transactions a year and provide both printed and graphic displays of actual v. budgeted expenses. Other features include a built-in database for tracking long-term assets and liability records, checkbook balancing and statements, and label file and year-end report generation. Price is $39.95.

TOTL Software, 1555 Third Avenue, Walnut Creek, CA 94596.

LEAVE IT TO BEEPER

No one is calling it artificial intelligence. That is still the stuff of which pulp novels are made. But the relatively new field of knowledge engineering is at least a baby step in that direction.

In Silicon Valley and across the country, knowledge engineers are attempting to discover the process by which experts in given fields make crucial decisions. These reasoning processes are then codified and programmed into computers. In this way, computers can be guided into simulating human inferential reasoning in specific fields of endeavor. A finished program is called an expert system. The advances have been made possible by the new breed of pow-



Chicks go crazy over home accounting programs like TOTL. Moneyminder.
READER SERVICE NO. 24

erful computers which can be programmed to solve problems using words and symbols, not only numbers.

According to a story by Robert Reinhold that appeared in *The New York Times,* knowledge engineering is currently being used to aid in prospecting for minerals, analyzing chemicals, configuring computers, and diagnosing diseases. New fields being pursued are military technology, investment banking, and genetic engineering. Major corporations such as Lockheed and Xerox are setting up their own knowledge engineering divisions, and knowledge engineering consulting firms are being established in a race for new clients.

Here's how it works. A knowledge engineer interviews experts in a particular field, trying to draw from each of them the methods by which they reach decisions and conclusions. According to the story in the *Times,* the experts bristle at first, maintaining that their most important decisions are based on intuition for the most part. However, in most cases, intuitive decisions are founded on reason and can be reduced to a series of "if-then" statements.

From this exhausting process of interviews of dozens of experts, hundreds, if not thousands, of "if-then" statements are derived, and a matrix of rules and connections forming the basis of decisions in that discipline is created. Since each rule or connection will not carry equal weight, each is given a numerical value. The information is all bound together in what is called a set of inference rules—defining the relationship between the evidence offered and the conclusions that can be drawn from it. Human reasoning is thus simulated when the computer "chains" through the rules: either reasoning toward a conclusion from given facts or backward from a desired result or solution toward conditions which would bring it about or evidence that would support it.

This "inference engine" is applied to data on a certain problem that the user inputs. After all the significant data is made available, the engine asks the user what questions he would ask to solve the problem. The engine then goes to work, scanning rules, sifting facts, and applying one to the other.

The engine, the expert system, only offers advice. It is ultimately up to the user and the executives peering over his shoulder to make the final decision. For now, the buck still stops here. That is, we think it does. We can't decide. The evidence points that way. A decision on the matter is pending.

SPRITE PROGRAM

SpryteByter provides C-64 users with over 60 commands for creating and manipulating sprites. You can create up to 184 sprites in one file and store them on disk or tape. Working sprites are displayed in all four shape modes (normal, expanded-x, expanded-y, and expanded-xy). Other features include movie mode (for animation review), data entry mode, and Game Maker utility for creating BASIC programs from sprite definition files. Control is via keyboard or joystick. The disk-based program retails for $34.95.

Microtechnic Solutions, P.O. Box 2940, New Haven, CT 06515 (phone: 203-389-8383).

NEW GAMES UPDATE

Activision, longtime lords of dedicated video games, have entered the C-64 market with two new games and four adaptations.

The Eastern influence is obvious in *Zenji,* which requires the

player to connect a glowing maze of elements (the Many) to a pulsating source (the One) to create a single unified green image, or Zenji. The puzzle/maze game is more difficult than it sounds, growing harder with each successful encounter.

Toy Bizarre is the latest assembly line game to roll off the assembly line. This one places your character in a toy factory gone berserk, where he must chase down escaped toys, turn off hissing valves, jump across assembly lines, and shun Hefty Hilda dolls. Safety checks, coffee breaks, and the timeclock add to the need for strategy.

H.E.R.O. seats you in a chopper which you must pilot to save miners trapped inside an active volcano. You can use your laser helmet to bore through granite walls, and dynamite charges to blast through impassable tunnels.

Beamrider pits a light warrior against enemy starships who have placed a shield around his planet. He must slice through all 99 layers of the sector and reopen the route to space.

The *Activision Decathlon* allows up to four players to compete in the ten Olympic events.

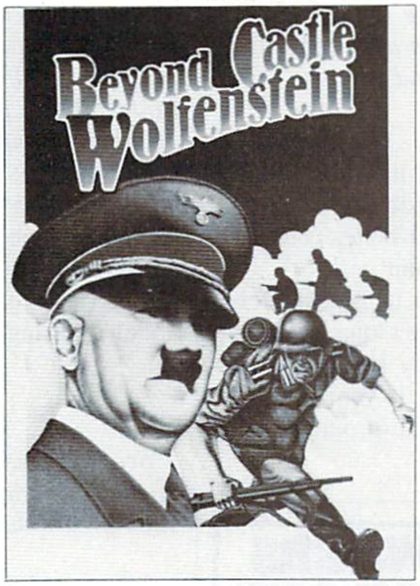
Pitfall, the vine-swinging classic, takes Pitfall Harry on a quest for jewels through a jungle swarming with dangers.

Price of each of these games is $34.95 for disk, $39.95 for cartridge. Coming 64 adaptations include *River Raid, Space Shuttle: A Journey Into Space,* and *Pitfall II: Lost Caverns.*

Activision, Inc., 2350 Bayshore Frontage Road, Mountain View, CA 94043 (phone: 415-960-0410).

Muse has announced *Beyond Castle Wolfenstein,* a sequel to the action/adventure game that follows a GI on his escape from a Nazi prison located in an ancient castle. Here, he must infiltrate Hitler's Berlin Bunker and assassinate the fuehrer. The sequel is a "stand alone," understandable and playable even by those who have not seen the original. On disk for the C-64; $34.95.

Muse Software, 347 N. Charles Street, Baltimore, MD 21201 (phone: 301-659-7212).



READER SERVICE NO. 42

From Infocom comes a new textual adventure titled *Seastalker.* This game has been written in the tradition of Jules Verne's *20,000 Leagues Under the Sea,* with the gamer in command of the underwater ship "Scimitar." Already old pros at interactive fiction software, the designers at Infocom have added a new angle to this game. They have designed it with the beginner in mind, as it can be played by children as young as nine. This is the first in a new series of games designed for this growing market. Another angle to *Seastalker:* it will ask for, and utilize, the gamer's name as the hero for the adventure.

Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138 (617-492-1031).

From CBS Software comes the educational *Webster: The Word Game.* For one or two children ages six and up, *Webster* will help children recognize familiar words, as well as assist them with spelling. Multiple skill levels allow the children to progress from one-syllable words to more complex terms. Available on cartridge or disk for the C-64.

CBS Software, 1 Fawcett Place, Greenwich, CT 06836.

Not wanting to miss out on the Commodore bandwagon, our northern cousins have begun issuing games for the C-64. Interphase Technologies has announced the release of *Viking Raider* and *Blockade Runner.*

Blockade Runner has the player as the commander of a space vessel attempting to transport supplies through enemy lines and asteroid belts to the home planet. This point-of-view game features multiple levels and fast action.

Viking Raider is Interphase's bid to put a halt to software piracy. The game is a diskette/cartridge combination: the cartridge simplifies interface with the computer, while the disk allows for substantial program storage. Thus, without the cartridge, a copy of the disk is useless. *Viking Raider* is an action-adventure game that combines graphics and interactive text.

Interphase Technologies, Inc., 6391-F Westminster Hwy., Richmond, B.C., V7C 4V4, Canada, (604) 270-8222.

From the shores of merry England comes a pair of new C-64 games. Bubble Bus Software has announced the release of *Flying Feathers* and *Bumping Buggies,* adding to their inventory of games. Address is 87 High Street, Kent TN9 1RX, England. Phone number is 0732-355962.

FLOTSAM

In our May issue we printed a letter from William Brauch of Canaan, VT, asking if we know of a genealogy program that would enable him to construct a family tree. Several readers responded to our plea for information, one of them being our own Richard Herring. He told us about Family Tree *by Genealogy Software, P.O. Box 1151, Port Huron, MI 48061 (phone: 519-344-3990). For the VIC or 64, the price is $39.95. For the 64 only, Richard informed us, there is* Family Roots *from Quinsept, P.O. Box 216, Lexington, MA 02173. Price is $185.00 plus $3.50 postage and handling. Thanks also to Vernon L. Whorl of Lexington Park, MD and Mrs. V. Wronski of Boston Bar, B.C.*

I just picked up your April issue, and wanted to write to commend you on the fine job you're doing. Each month you guys get better and better. There are a couple of things I'd like to point out:

1) The article on *Easy Script*, and its quick reference sheet, was excellent; only there's one command that Morton Kevelson left out. I only know about it because I discovered it by accident. If you type fl CONTROL/3 or fl CONTROL/f, the program will play a little song for you.

2) My copy of Bug Repellent works fine, except that if I print it to the screen, I can't get it to stop, and lose the last few line numbers. This means that I have to output to the printer every time. Can't something be done about this?

Brad Morris
Waterloo, IA

Thanks for the information about Easy Script, *Brad. As for your problem with the Bug Repellent, we can't overemphasize the fact that you must read the introduction to each of our programs before typing them in! If you'll refer to the short program introduction on page 69 of this issue, you'll find the solution to your problem.*

I just found *Ahoy!* on the newsstand, and was happy to see such an excellent magazine on the market.

Payment for a one-year subscription is enclosed. I'd also like to get all the back issues. How do I go about ordering them?

Stephen Marhoffer
Union, NJ

See our back issues ad on page 36. Our first issue is sold out, but we're offering photocopies of that issue's programs and feature articles for $4.00.

I just read the interesting article on spreadsheets (May '84) and found it very helpful. As a new computer owner, I've found that one of the worst areas of marketing and information by computer companies and the Media is what to do with the blessed things.

In the six months of reading I've done I have found nothing really useful to my 64 for around the house. Lots of games and some "make work" programs. For example, keeping a computer checkbook is very possible, but more time-consuming than to do it the old-fashioned way on paper.

Now, my wife is going to open her own business, and I'm trying to figure out how we can use the 64 to help us. I can think of a few applications (mailing lists, accounts payable, etc.) but do not ever recall seeing an article in any magazines about practical applications for a computer in a small business. I would like to see one that lists the uses, and tells what type of software would be necessary.

Does a spreadsheet do it all, or do I need a database as well? The same kind of article would be useful for the home as well.

Richard Peterson
Bedford, TX

WHAT'S MY JOB?



By B.W. Behling

Perhaps you recall a T.V. show of a similar name that was popular in the late 60's? This is a computerized version of that show, with your C-64 playing the part of the panel of experts. It is a game that will exercise your mind rather than your joystick hand.

The game starts by knowing two occupations and learns more as you proceed. When the program is first run you will be asked if you want to load all previous jobs. If this is the first time you are using the game, answer "NO" or "N" and continue on; we'll come back to this later.

The next thing you will see is the main board listing the instructions for the game and asking if you are ready to continue. At this point you can list all the jobs it knows or you can continue on to the game proper. If you type "L" all the jobs it knows will be listed. Typing "Y" will call up your first question. As the game proceeds it will try to guess the occupation you are thinking of by asking short questions that you answer with a "Y" or "N". If the game asks a question that is not a definite "Y" or "N" answer you should answer with whatever is most correct, i.e.:

> "Do you work mostly outdoors?" For a model, answer "N". (Although some work outdoors, most work in a studio.)

After the game has exhausted its supply of questions it will make a guess based on the answers you have provided. If its guess is correct it will ask for another job and start over. If its guess is incorrect it will give up and ask for the job you were thinking of. Next it will ask you to supply a question to differentiate between its guess and your answer. In the beginning you should use broad general questions and later move up to more specific questions, i.e.:

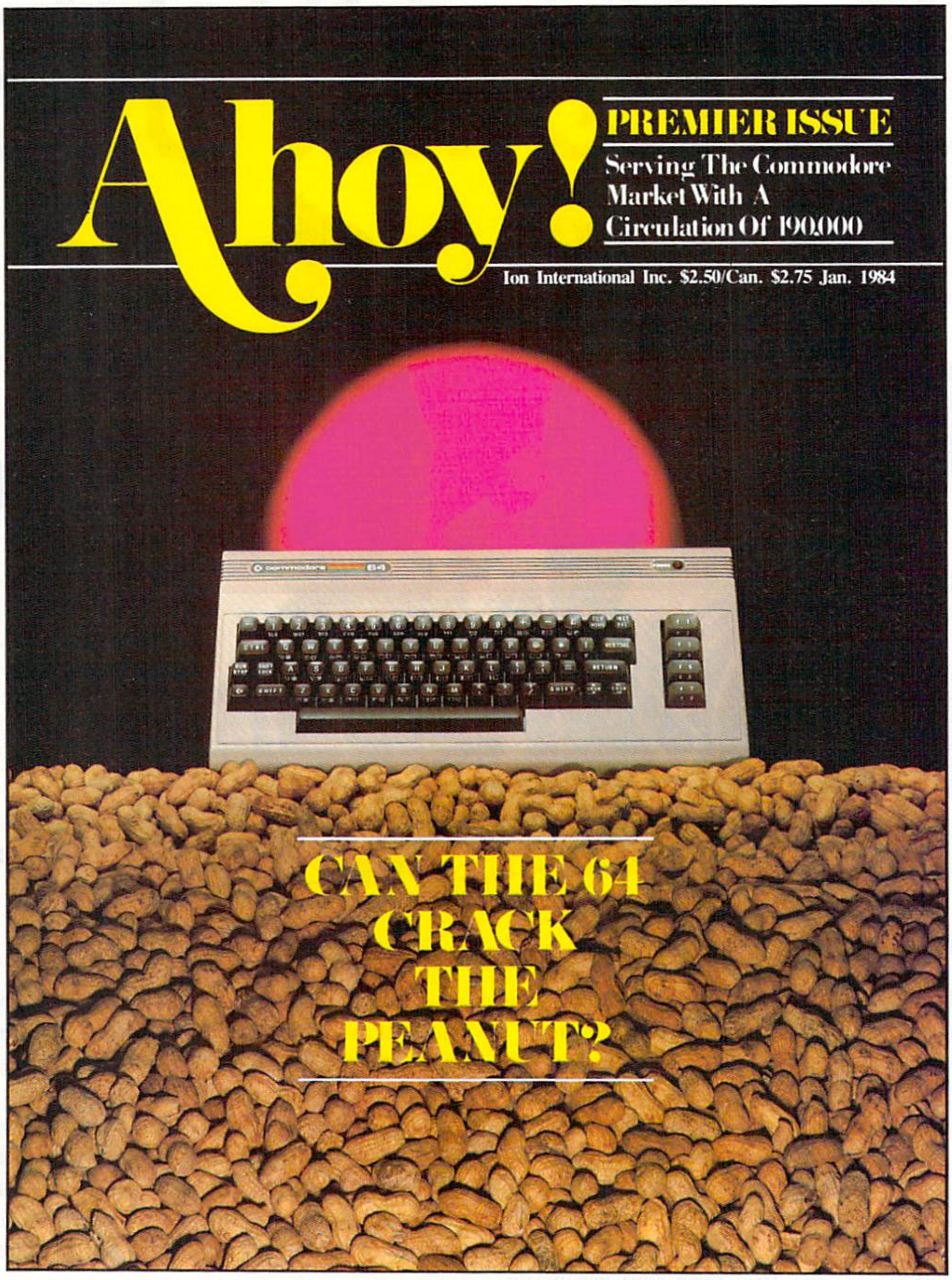
Beginning:
- Computer's guess: FARMER
- Your job: POLICE OFFICER
- Sample question: Do you wear a uniform?

Later on:
- Computer's guess: DOCTOR
- Your job: NURSE
- Sample question: Do you assist doctors?

Jobs should be no longer than 19 letters and spaces and questions should be no longer than 38 letters and spaces. The program will automatically add the appropriate question marks, so these should be left off your questions.

Jobs and questions, once entered, are kept in variable A$(0-2000) and are saved in file #1"Jobs A$" upon command at the end of the game. I would rec-

Continued on page 23



HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $19.95 ($26.95 Canada and elsewhere)
☐ Twenty-four Issues for $37.95 ($49.95 Canada and elsewhere)

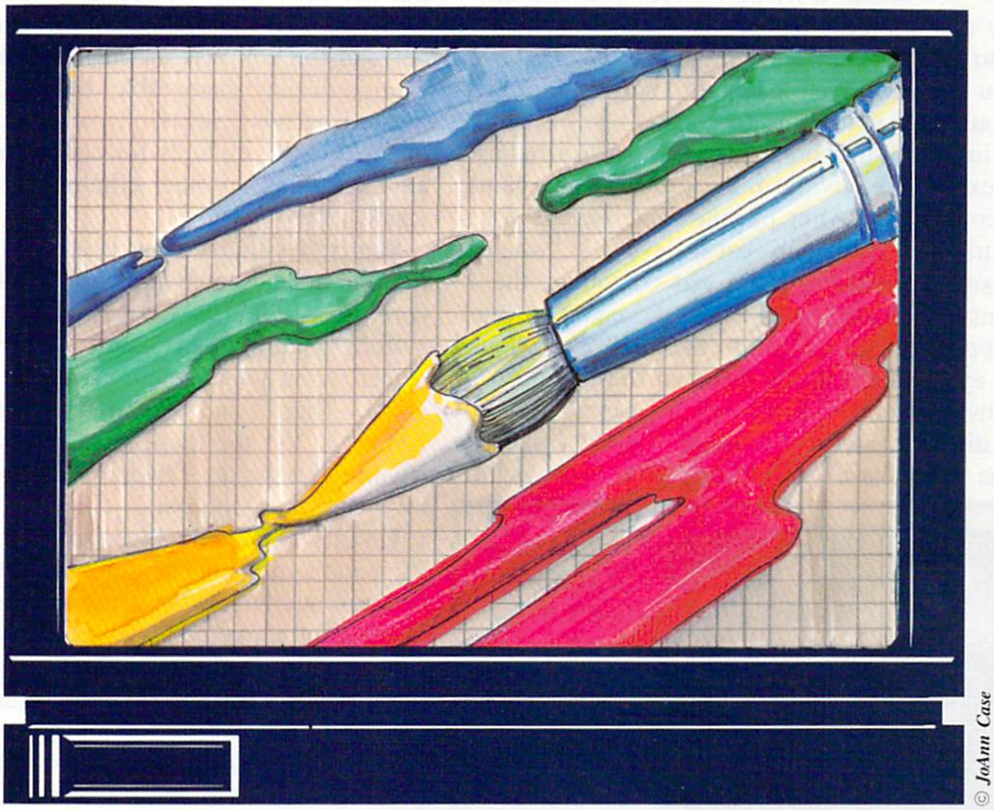
Name____

Address______________________________________

City____________State____________Zip____________

Send coupon or facsimile to:
ION INTERNATIONAL INC.
45 West 34th Street, Room 407, New York, NY 10001

CREATING YOUR OWN



© JoAnn Case

GAMES ON THE VIC AND 64

PART II: THE WORLD OF THE GAME

By Orson Scott Card

WILL PLAYERS WANT TO STAY IN THE WORLD THAT YOU CREATE FOR THEM? THE SCREEN DISPLAY IS A VITAL PART OF ANY ACTION GAME THAT TELLS A STORY.

Arcade-style games have a lot of things in common with a movie or a book. One of the most important is that games can create a world that the player lives in while he's playing—and, sometimes, for long afterward.

You've probably read a book or seen a movie that you wished would never end. Tolkien's *Lord of the Rings* affects a lot of people that way—they even read the appendix at the end of the book, just so it won't be over. *Star Trek* did it for others—the endless reruns have their audience. Storytellers can create worlds for people that they could never live in otherwise.

And computer games have added a whole new dimension to that. You don't just read about what other people did, or watch other people act their way through a movie. You actually control one of the actors. You can decide what happens next.

A good game designer can give you experiences you won't ever have in the real world. For instance,

Joust lets you mount a flying ostrich; you rise into flight by making the bird flap its wings, and then enter into combat with other mounted enemies. Not what you expect to have happen tomorrow morning when you finish your Wheaties, is it?

Even in games that don't give you such a radically new experience, living in the milieu of the game can be compelling. After playing *Lode Runner* for four hours (and I still only got to level 93), when I read a book I found my brain was continuing the game. Instead of reading along to the end of a line, then skipping left and down to the next one, I found that my eyes wanted to dig holes in the line and jump down, or climb ladders up the page. It made reading difficult, and it also made it clear to me that the game experience has staying power in the mind.



The concept of* Tempest *is abstract, but the game's shapes and patterns have a compelling visual rhythm.

There are many ways a game world can have such powerful effects on the players who dwell in it. The sheer beauty or realism of a dazzling screen display can do it. Novelty helps, too—players are obviously more excited when a game shows them something they haven't seen before. If the display is active, so much the better—the world will feel more complete if things are happening. It is not just "bells and whistles" when the visual display is animated, realistic, complete, even when these elements don't pertain directly to the action.

WHERE DO GREAT DISPLAYS COME FROM?

Stunning screen displays don't come out of nowhere. There must be a strong concept. Sometimes the concept is obvious. In a football simulation, for instance, the game itself determines the general outlines of the display. But game designers can use the same basic idea in wildly different ways.

Sometimes the story of the game suggests the shape it should take. The broad outlines of previous climbing games gave *Donkey Kong* its basic shape—horizontal surfaces, ladders, obstacles to jump over. But the story of the gorilla that carried off Mario's girlfriend and carried her upward into a building under construction was the idea that made *Donkey Kong*'s display memorable. Steel girders, scaffolding, oil barrels, elevators, conveyor belts—these opened up possibilities for action, obstacles, and animation that no other climbing game could match at the time.

Sometimes the concept is abstract, purely visual, like the arcade game *Tempest,* in which the player controls a figure that moves around the perimeter of shifting geometric shapes, firing downward at the center, from which enemy shapes arise in a deadly attack. There is no real world situation even remotely like it. Yet the shapes and patterns have a visual rhythm that makes *Tempest* a compelling game just to watch, let alone play.

Ultimately, memorable displays come out of someone's mind. Unimaginative games are created by programmers, whose only concern is the creation of tight, effective code. There is a beauty in good programming, but players don't experience the code—they experience the sights, sounds, and actions that come from the television set. Good paintings come from artists who see beyond the techniques of perspective, chiaroscuro, and color, who have a clear vision of what the painting *ought* to be, and use their techniques only as a means to that end. Likewise, good visual displays in computer games come from designers who regard programming techniques merely as a means to an end, not an end in themselves.

In other words, if you're going to create a really memorable game, it won't be enough just to learn programming techniques. You have to think of good things to do once you've mastered them. In fact, a really good game idea can often lead you to solve programming problems or develop programming techniques that other people have not thought of yet.

It works both ways, though. As you learn the methods of programming, new possibilities occur to you. You begin to think, with this technique I can do *this* or *that,* and a humdrum display suddenly takes on life and power.

Does it sound mystical and magic? In a way, it is. Game programming isn't math or logic, it's art. Or rather, it's math and logic used as the tools to create art. (And you thought you were just manipulating loops, branches, variables, and vectors!)

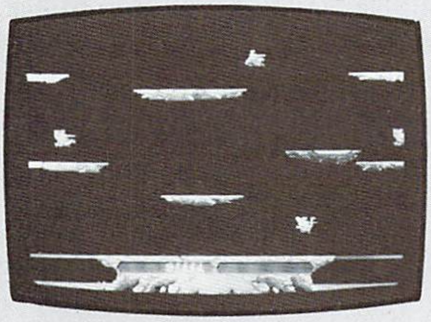
PRINTING A RANDOM SCREEN

Many games keep interest alive by setting up a new screen every time you play. A randomly gener-

ated display must be carefully planned, however, so that the game is still playable.

The basic technique is to set up a group of display elements, like pieces in a board game. Each element will be either a single character or a complex character stored in a string variable. The largest piece in our example program will be an ocean liner, four characters wide. The other movable pieces are a cruiser, two characters wide, and a quick little speedboat, which consists of a single character. Then we'll have small stationary buoys, one character in size, and lighthouses, which consist of two characters stacked one atop the other.

(To demonstrate the methods of setting up a random screen, I could ask you just to imagine the shapes. But little groups of letters and numbers moving around the screen aren't very satisfying. So I'm including a set of seventeen custom characters and the subroutine to set up a custom character set. I won't even attempt to explain what's going on, and you can leave out these lines—500 through 510 and 1000 on. It will save you half the typing in this program—but you'll miss most of the fun of the display. You must have a line 500, but it can be 500 REM.)



***Joust** provides a gamer with experiences unlikely to occur in real life. READER SERVICE NO. 25*

FITTING THE DISPLAY TO THE GAME

Let's keep the game concept simple. The player controls a small boat that starts at the bottom of the screen. A current is irresistibly moving the boat up the screen toward the top. It will drift upward, no matter what the player does. However, the player can move the boat slowly from side to side.

As the boat drifts up the screen, it must avoid various obstacles. There are large and small boats docked in the river. There are also lighthouses. A collision with a lighthouse is total disaster—the boat sinks. A collision with a ship can be a serious problem, depending on the ship's size. Collisions with the small buoys in the water cost only a few points. However, if a ship is colored *green*, the player's boat needs to dock with it in order to exchange cargo—there are lots of points for touching each green ship, and even more for making it to the top of the screen.

When the boat reaches the top of the screen, it starts over again at the bottom—but with a new, randomly generated arrangement of ships, lighthouses, and buoys. This means that, for the game to work, we must come up with a new screen display every few minutes.

In setting up a program to generate the screen display, we have to keep several things in mind:

1. We don't want to PRINT any ships or lighthouses or buoys on top of or touching each other.
2. We want to have some random areas of the screen blank—a good mix of the different ships facing in different directions, lighthouses, buoys, and blanks.
3. We want the screen to *look* random—things shouldn't be lined up in neat rows and columns like an orchard.
4. We want lots of different colors.

Here is a plan that will enable us to meet these requirements:

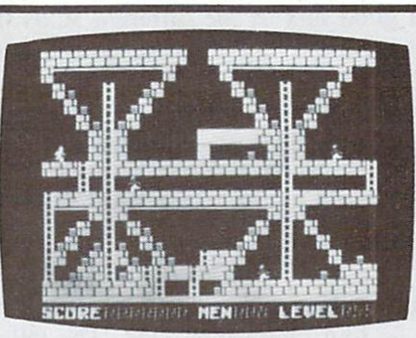
Divide the screen into cells. The widest figure, the ocean liner, is four characters wide. The tallest figure, the lighthouse, is two characters high. In order to leave at least one blank space above, below, and beside every figure, we'll think of the screen as if it were divided into little boxes, or cells, six characters wide and three characters high.

Random numbers. Our routine must randomly decide which figure to put in each cell—or whether to put a figure in at all. Also, we must have another way to randomly decide where within the cell to PRINT the figure. With cells six spaces wide, even the ocean liners have one space leeway from side to side, and the narrowest objects—buoys, lighthouses, and speedboats—could be in any of five different positions within the cell. We need this random placement within the cell because if every figure were PRINTed at the lefthand edge of its cell, we'd get a neat all-in-a-row appearance. Another random routine must decide whether to have a ship face left or right. And, finally, we must have a random number to decide what color the ship should be. (Lighthouses will always be white; buoys will always be black.)

This means we need to generate four random numbers and two index numbers. The index variable *I* will set the row number—we'll use every third row. The index variable *J* will set the cell number—we'll use every sixth column to start a cell. We'll

generate a random number *K* to determine if a cell will be blank or, if it is not blank, which figure it will contain. We'll use the random number *L* to set the position of the figure within a cell, the random number *M* to choose among five possible colors, and the random number *N* to choose which direction the figure will be facing.

Set up the building blocks. The figures that we put into the cells are like building blocks, which are set up in a subroutine executed once, at the beginning of the program. The three ships are stored in the string array SH$*(x,y)*. Each ship will have two possible shapes, one for moving left, the other for moving right. The first index in the array (the *x* index) will determine which of the three ships is displayed; the second index *(y)* determines which direction the ship will face. If the *x* index is 0, the ocean liner will be displayed; 1 is the cruiser, and 2 is the speedboat. If the *y* index is 0, the ship will face left; if 1, then right.



Living in the milieu of a game like* Lode Runner *can have lasting effects.* READER SERVICE *NO. 26

This is done in lines 520 and 530 of the program:

```
520 SH$(0,0)="@ABC":SH$(0,1)="DEF
G":SH$(1,0)="HI":SH$(1,1)="JK"
530 SH$(2,0)="L":SH$(2,1)="M":BU$
="N":LH$="O"+CHR$(17)+CHR$(157)+"
P"
```

GENERATING THE RANDOM DISPLAY

In the program *Random Display* on page 86, lines 640 to 670 contain the screen generating subroutine. In addition, lines 700, 710, 720, 730, and 740 each contain sub-subroutines to PRINT one of the figures within a cell. Let's look at those routines closely here. (If you remember last month's column, we also use the variable BE to contain the number of the last usable line at the bottom edge of the screen, and the variable RE to contain the number of the last usable column at the right edge of the screen.)

```
640 PRINT CHR$(147):FOR I=0 TO IN
T(BE/3):N=INT(RND(9)*2):FOR J=0 T
O INT(RE/6)
645 M=INT(5*RND(9))
650 K=1+INT(RND(9)*10):ON K GOSUB
 700,670,710,670,720,670,730,670,
740,670
660 NEXT:NEXT:RETURN
670 RETURN
```

In line 640, we clear the screen—CHR$(147)—and then begin two nested loops. The outer loop sets the variable *I* to every third line, starting at the top of the screen (0) and ending before the bottom line (BE). The inner loop sets the variable *J* to every sixth column starting at the left edge of the screen (0) and ending before the right edge (RE). Within the line loop (I), but outside the cell loop (J), we generate the random number *N*. This means that every figure in a given row will face the same direction, but that any row can face either left or right.

Notice that the nested loops mean that we will start at the upper left-hand cell, deal with each cell in the top row, and then jump down to the third row and do each cell in that row from left to right, until we reach the bottom right-hand corner of the screen.

Line 645 sets the variable *M* to a number between 0 and 4. This number will be an index into the color-assignment array, CL$*(x)*, which has been previously set up so that each element in the array contains a different color character. When PRINTed, each color character causes all other characters PRINTed afterward to be the chosen color. It's just as if you pressed the CTRL-color or Commodore-color key combinations.

Line 650 sets *K* to a random number from 1 to 10. Then the ON statement uses *K* to jump to one of ten subroutine lines. Notice that half the subroutine lines are 670, and line 670 does nothing except RETURN. This means that approximately half the time, the ON statement will do nothing—leaving approximately half the cells on the screen blank. The other five lines lead to subroutines that PRINT one of the five possible figures on the screen. This means that if any figure is PRINTed in the cell at all, it has an equal chance of being any of the five figures.

What if we wanted many more buoys than any other figure? We could adjust the chances by replacing line 650 with these two lines:

```
650 K=1+INT(RND(9)*16)
```

```
655 ON K GOSUB 670,670,670,670,67
0,670,670,670,730,730,730,730,710
,700,720,740
```

About half the cells will still be blank, but now a quarter of the cells will contain buoys, and the other four figures will fairly evenly divide the remaining fourth of the cells.

Line 660 closes both loops and, when the whole screen is set up, returns from the subroutine. Line 670 is the empty subroutine that causes a cell to remain blank.

Here are the five routines that actually PRINT the figures in the cells. Each routine ends with a RETURN statement.

```
700 PRINT VP$(I*3)TAB(6*J)CL$(M)S
H$(0,N):RETURN
710 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*4)
715 PRINT VP$(I*3)TAB(6*J+L)CL$(M
)SH$(1,N):RETURN
720 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
725 PRINT VP$(I*3)TAB(6*J+L)CL$(M
)SH$(2,N):RETURN
730 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
735 PRINT VP$(I*3)TAB(6*J+L)CHR$(
144)BU$:RETURN
740 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
745 PRINT VP$(I*3)TAB(6*J+L)CHR$(
5)LH$:RETURN
```

Because there isn't room within a cell to move the ocean liner around, the routine at 700 is the simplest. It finds the cell by PRINTing cursor-down character contained in the string VP$*(x)*—the right row is found by multiplying I*3. Then it TABs in to the right column number for the cell—J*6. It PRINTs the color—CL$(M)—and the ship itself—SH$(0,N)—with N determining which way the ship is facing.

Each of the other routines takes two lines. The first line of each pair sets *L* to 0. Only if the cell does not extend beyond the right-hand edge of the screen is *L* set to a random number. When the figure is PRINTed in the next line, the random number *L* is added to the TAB number to place the figure in its random position within the cell.

That's all. Each time the subroutine at 640 is called, the whole screen is cleared and then set up randomly, but according to strict rules that make sure the screen display works.

The rest of the program is pretty well self-explanatory. Line 10 routes the program to the setup routines. Lines 100 to 120 call the screen generating subroutine, then wait until any key is pressed before creating a new screen. You can keep generating new screens by continuing to press any regular key. To stop the program, press RUN/STOP-RESTORE.

Lines 600 to 620 set up the vertical movement strings (HOME and cursor-down characters) and the color array CL$*(x)*.

LAUNCHING THE SHIPS

The random screen is fine—but those ships shouldn't have to stay at anchor all the time. Let's launch them, get them moving.

We'll arrange the screen like this: At the top there'll be a row of buoys, lighthouses randomly arranged. These won't move. Then there'll be a few blank rows in which the ocean liner will sail back and forth. Then another row of buoys and lighthouses, then some rows where the cruiser will move—only twice as fast as the ocean liner. More buoys and lighthouses, and then a space where the speedboat will zip along twice as fast as the cruiser and four times as fast as the ocean liner. Finally, near the bottom of the screen there'll be one last row of buoys and lighthouses.

Most of this program, *Moving Display,* is the same as the *Random Display* program, but to avoid confusion, both programs will be printed in full on pages 86 and 87.

The screen is still generated randomly, but now the only figures that will be put on the screen at first are buoys and lighthouses, and instead of cells six columns wide, the cells will be only two columns wide. The lines from 640 to 710 are changed accordingly. When executed, this routine puts four rows of buoys and lighthouses on the screen.

Another change is in the ship's shapes, which are assigned in lines 520 and 530. Now, any ship that will move left has a blank space added to the right side of the figure, and any ship that will move right has a blank space added to the left side. This means that as the ships move along, they will automatically erase one space behind them—we don't have to PRINT a blank to erase the ship at its old position. This means there'll be no blinking when the ships move.

Since the long ocean liner will reach the right-hand edge of the screen sooner than the smaller ships, and the cruiser sooner than the speedboat, each of them must have its own right-hand edge

Continued on page 94

BOOK REVIEW

THE WORKING COMMODORE 64 by David Lawrence (Creative Computing Press, 1984; $9.95). 176 pages, softbound.

The Working Commodore 64 is a programmer's delight! Nonprogrammers can enter its programs for their general utility, but the programmer can harvest it for its subroutines and programming ideas. Beginners may find this collection of 20 programs rough going. All other skill levels will find valuable routines. The book is a treasure house of useful techniques which can be adapted to a wide variety of programs.

All programs are presented in modules, many of which can stand alone or be transported to other programs without change. The reader can enter each module, most of which are short, and test it before appending it to other routines which comprise the complete program.

The author provides a commentary on each module, relating it to the other parts of the program. Line-by-line annotations and instructions for testing the module complete each subsection. The annotations are full of gems. For example, in the "Artist" program, Lawrence demonstrates an easy technique for accessing all 16 of the 64's colors without having to use the Commodore key.

Reading a list of the complete programs doesn't indicate how useful this book is. The complete programs include a text editor, a drawing routine, a mini-database, some educational drills, and some accounting programs. Each is a meaty, useful program of its type, and can be put to work with no programming changes.

The modules within each program, however, can be mixed and matched to produce an unlimited number of special applications. For example, the "Artist" is a drawing program. It has four routines. The first, "Variables," simply sets aside room for initializing variables. The second, "Cursor, Move, Print," is the heart of the program, and can be used by itself to draw on the screen. The third and fourth, "Save Design" and "Save Screen," permit the artist to save a single design or a whole screen.

Lawrence covers all points. He includes routines for business and pleasure graphics, music, business and home applications, and educational software. He offers programs for merging, deleting blocks of lines, and renumbering programs to facilitate using the modules to make complete programs.

Program listings are fairly clear for dot-matrix reproductions. The book begins with the best key for deciphering the graphic symbols in Commodore programs that I have seen. All programs save to tape, so disk drive owners will have to write their own I/O routines.

Almost all the programs use memory locations specific to the Commodore 64. VIC owners can probably use the book, but will have to adapt. The programs use the Commodore graphic control symbols heavily. They are not readily transportable to other machines.

Lawrence says in his preface that he wanted to write "a solid collection of programs that would be worth having." He succeeded very well. At $10, *The Working Commodore 64* is an irresistible bargain.

—Annette Hinshaw

CHECKLIST

for the C-64

By Bob Spirko

Everyone makes lists. As a hobbyist, shopper, or whatever, you often find yourself drawing one up. And, at times, that uneasy feeling arises as you wonder if you have left something out.

But there's no more need to worry. *Checklist* is very easy to use, flexible, and safeguarded against miss-keying. You will find the screen display neat and attractive through the use of extended background colour mode.

As an avid tour cyclist, I wrote *Checklist* as an aid in selecting equipment for any given trip. My equipment list will vary depending on the trip's length, its degree of difficulty, and the weather (not to mention whims). Moreover, as I am an ultralight enthusiast, I find it of special interest to keep track of my load weight. With *Checklist,* I am able to list all items and their weights in any category (clothing, food, etc.). Then, category by category, I go through my base list and choose the items that I would like to take. I may also add any items not on the base list, and I can make changes in any item (if I want to take three pairs of socks, then I adjust the weight accordingly). As well, I can code items, so that if I have two similar items I can distinguish between the two. Also, of course, I can strike an item off my list, should I change my mind.

While all this is happening, a record of the total weight and category weight is displayed. Once I have gone through all the categories, the screen displays the categories, subtotals, and total weight. From there I may have a list of the selected items displayed, or go back to any category and make changes, or produce a hard copy of my selections.

But you need not be a cyclist to make use of *Checklist.* Just select your own DATA. DATA statements are formatted in the following manner (the program includes a few DATA lines for demonstration purposes):

```
DATA A1$,B1$C1$,D1,B2$C2$,D2,....,
Bn$Cn$,Dn,0,0
```

where A1$ is the category name, B1$ is the first item, C1$ is the code and D1 is the weight (if you wish to use a different standard such as dollars, then change lines 33 and 2510). Category and item strings may have a length of up to 12 characters. The code is a single character and D2 is a four digit number (three digits if it contains a decimal). The items for a category are READ until it encounters a 0,0 and then it looks for a new category name. It stops READing when it encounters END. You may have up to 22 items in a category, as it is limited to screen length. But you can have more items under the same category by creating another category under the same name.

When displayed on the screen, the items are listed with an alphabetic label (e.g., the first item is A) and selections or deletions are made by pressing the respective key. All selections are accompanied by a pleasant tone. Pressing the wrong key results in a buzz sound.

The following explains the use of the keys:

f1-DEL. This will delete any item that you have selected and adjust the weights accordingly.

f2-CHNG. After you have gone through the categories and made your selections, you may wish to go back to make changes. After pressing f2, a category list is shown and you will be asked for the category. The category listing will appear as you left it and you can effect the necessary changes.

f3-EXTR. When you wish to add another item not

on the base list, use this key. You will be asked for the name of the item and its weight. It is then coded with an X and added to the end of the list. If you wish to change the code use f5.

f4-TOTL. This takes you to the "page" that displays the subtotals and total.

f5-NEW. Here you will be asked for the item that you wish to change. After this, you can enter its new weight and code. If you do not wish to enter a new code, press RETURN. It will not accept the item that you have already selected. If you wish to change such, then simply delete the item, change the weight or code, and place it back on your list.

f6-COPY. This produces a hard copy of your selections along with all the weights. Each item is listed with a check box for convenience.

f7-NEXT. Pressing f7 allows you to proceed to the next category listing.

f8-LIST. A list of all the items that you chose (up to 69 items) is displayed.

E-END

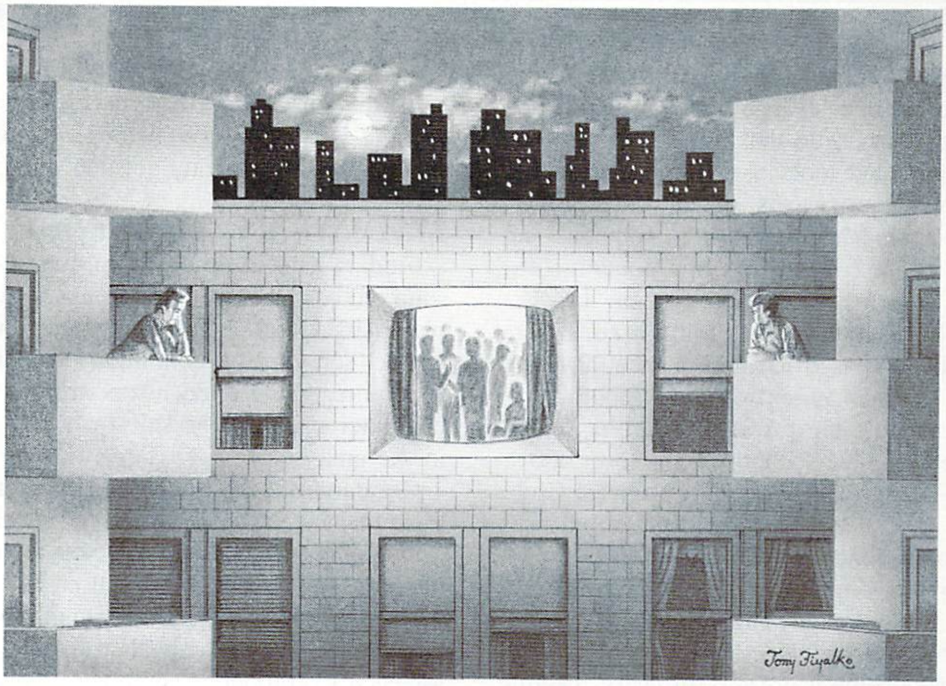
Each page will display the keys that can be used at that time. If you enter either the DEL, NEW, or EXTR modes accidentally, you can escape by pressing RETURN.

The program lines and principle variables are:

Lines 115-260:	Select item
Lines 400-480:	Check selected item on list
Lines 500-560:	List items in category
Lines 800-820:	Print total
Lines 1000-1110:	Convert number NO to character string with length P
Lines 1100-1110:	Convert string to upper case (for background colour)
Lines 2000-2010:	Beep
Lines 2100-2110:	Buzz
Lines 2500-2590:	Extra item
Lines 3000-3090:	Delete item
Lines 4000-4170:	List categories, subtotals, and total
Lines 5000-5099:	List selected items
Lines 6000-6180:	Make changes in any category
Lines 7000-7170:	Print hard copy
I$(X,Y):	Y item in category X
G$(X,Y):	Weight of item
TI$(X,Y):	Selected item
TG$(X,Y):	Weight of selected item
N(X):	Number of items in category X
S(X):	Total weight of selected items in category X
M:	Number of categories

SEE PROGRAM LISTING ON PAGE 88

STARTING OR JOINING A USERS GROUP



By Joycelyn Woods

Prior to 1983, the Year of the Computer, owners of personal computers tended to be hobbyists, engineers, or individuals with some knowledge of programming. Such users didn't mind tinkering, especially since microcomputing was looked upon as a hobby.

But thanks to the VIC 20 and C-64, practically everyone can own a computer today. And the newer users are more likely to look upon their micros as appliances—their main interest being, what can their computers do for them? But to the newly initiated, even the most user-friendly system can appear ominous.

Today's users are finding, just as yesterday's did, that the best place to obtain information is from one another; and that the most convenient forum for this exchange of information is a users group. Five years ago, most users groups were generalized: it didn't matter what brand of computer you owned, or what your interests were. Such groups still exist in remote areas, but most of the earlier groups have grown so large that they've broken up into sub-groups of computer brand or specific interest.

One thing hasn't changed: because of the lack of user support, a users group is still the best way to learn more about your computer.

WHY A USERS GROUP?

To illustrate by example some of the benefits of belonging to a users group, I'll run down some of the activities of my own—the New York City VIC 20/C-64 User Group (NYCUG). It started about two years ago, when 10 or 15 people began meeting as the Commodore sub-group of the New York Amateur Computer Club (NYACC).

With the rise of the C-64, our group grew quickly—and so did our enthusiasm. We began inviting companies to give demonstrations. The Alien Group delighted us with their singing Voice Box. Micro Ware came up from New Jersey and demonstrated the Connection interface, along with the rest of their line.

Most users groups, of course, are not located in areas like New York City, with a number of hardware/software manufacturers nearby. But the fact is that the demonstrations by our own members have proved even more interesting. Our resident artist, who had done the sample demos that come with the *Doodle* graphics package, amazed us with his work one evening. Our Sysop brought his local bulletin board down so that members without modems could see what telecommunications was like, and what they were missing! We've had lectures on FORTH,

beginning machine language, simple BASIC arrays and loops. One member even brought in his disk drive which had burned up that week so that we could take it apart and look inside. We've had demonstrations of word processors, databases, terminal programs, and spreadsheets. Recently, someone suggested that we begin collecting public domain software; we now have a librarian working on it.

When the CP/M cartridge was introduced, several members who'd purchased it came to meetings complaining about the lack of software to support it. This initiated our first big project, which has been to create a public domain CP/M library of over 100 disks from the New York Amateur Computer Club's CP/M library. (Those interested in purchasing disks should write to Commodore CPM/NYCUG, c/o Amateur Computer Club, P.O. Box 106, Church Street Station, New York, NY 10008.)

But by far the most valuable benefit of a users group is the friendships it creates. In every meeting of every group I've attended, I've found the members helpful, even selfless in solving a problem for someone.

I hope you now have some idea of what you're missing if you've never belonged to a users group. Your next step is clear: get into one.

FINDING A USERS GROUP

Finding a local users group may require some resourcefulness. Since most groups require a meeting place larger than a member's living room, you could contact schools and community centers in your area to inquire. Quite often, users groups will meet at a computer store, or the salesman may know of groups in the area.

Commodore publishes a list of users groups in their magazines, *Power/Play* and *Commodore: The Microcomputer Magazine.* If you or a friend own a modem, you can access the user group directory in the Commodore Network through Compuserve. (The Network's 1000+ pages also include a bulletin board for each Commodore computer, a conferencing area, a hotline for questions, articles from the Commodore publications, tips, an extensive database for down-

loading programs, and listings of Commodore dealers.) Log onto Compuserve in the usual way, then at any ! prompt type GO CBM 248 and hit RETURN. This will place you at the users group directory.

NATIONAL GROUPS

If you have no group in your area, you could join one of the larger national groups—sacrificing, of course, the important interchange of ideas.

The Toronto Pet Users Group is one of the oldest and largest national organizations, boasting over 14,000 members (11,000 of them outside the Toronto area). An associate membership for someone not in the Toronto area costs $20.00; if 15 or more join, the price drops to $15.00 per person. This includes 10 issues of *TPUG Magazine* and access to their library of over 200 disks. A disk of programs which goes with the magazine can be purchased for $10.00 per month; or, if you join with a group of 25 members or more, you can get one free. (40-member groups get two monthly disks free, and so on, up to a maximum of six disks for a group of 240 members.)

TPUG maintains an extensive list of users groups, some as far away as New Zealand, that is updated weekly. In the Toronto area, the group holds several meetings each month and offers lectures and conferences for members. Address is 381 Lawrence Avenue West, Toronto, Ontario, Canada M5M 1B9.

The Commodore Club has been in existence for slightly over one year. The cost to join is $10.00, which includes a monthly newsletter with tips and short articles, and discounts on Commodore products. Their address is 1200 Wilson Drive, West Chester, PA 19380.

STARTING A USERS GROUP

If you can't find a local group but want the personal interaction that a national organization can't provide, you'll have to start your own users group. It can be as simple as five or six people meeting once a month to exchange ideas. (But chances are it will grow, so be prepared.) Starting up requires some work—you'll need a few people to help out. Begin by selecting a temporary president and secretary to serve until all the officers can be elected. Next, select a name. It's best to use one that identifies you. You could include Users Group (UG) or Interest Group (IG) at the end; if you use a computer name, it must be the trademarked name.

A constitution and bylaws should be drafted. It's best to keep these simple. You can get a book on drafting a constitution for small organizations at the public library. The stated purpose of the organization (Article II) should be "to promote computer education for the Commodore computers." Once the constitution is drawn up, you can elect your officers and vote to accept the constitution and bylaws.

You are now a users group. So that others in your area can find you, send your group's name, address, contact person, and telephone number to *Ahoy!* so the information can be published in this column. You can send the same information to the Commodore Club and TPUG, as well as to many of the other computer magazines. Don't neglect to post messages locally, at supermarkets, schools, etc.

As your group membership grows, you can tackle projects like setting up a bulletin board, or putting together a library of public domain software. (Remember: using copyrighted software for your library is against the law.)

In selected future issues of *Ahoy!*, a listing of users groups will appear. *Ahoy!* has realized from the beginning the valuable service that users groups can provide for the Commodore community, and will continue to provide information of specific interest to users groups in this column. If your group has done anything noteworthy, tell us about it so we can publicize it in a future installment. □

WHAT'S MY JOB?

Continued from page 11

ommend that you use a separate tape for this file, as it will make saving and loading it that much easier.

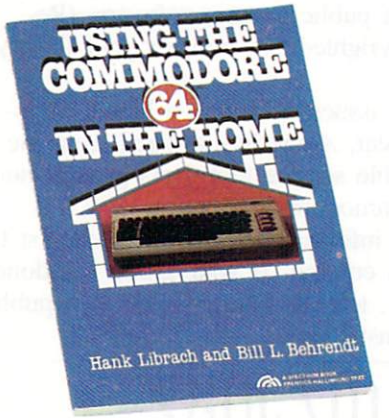
To save your job file you should type "N" when asked if you are ready and then insert the tape you are using for that purpose, type "Y" and then "*" and press record and play on your Datasette. (Disk drive owners: although written for tape use, the program may be easily adapted to disk use. See your users guide for details.)

Loading in the job file is a simple reversal of the above procedure, involving following the instructions given when you type "Y" to the question: "Do you want to enter all previous jobs?" at the start of the game. One last note before I go: when new jobs are to be saved in the tape file, you should record directly over the old file to avoid errors. The old jobs will be saved again along with the new entries.

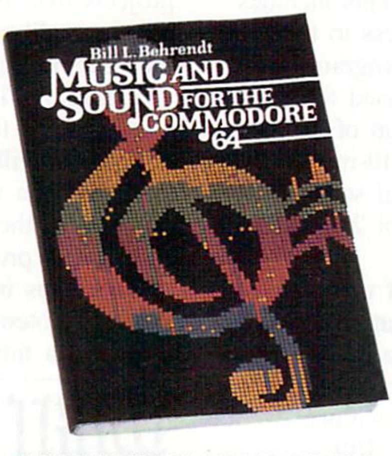
There you have it. Keep reading *Ahoy!* for more valuable programs—such as one I'll be providing in an upcoming issue that will turn your monitor into an aquarium! □

SEE PROGRAM LISTING ON PAGE 70

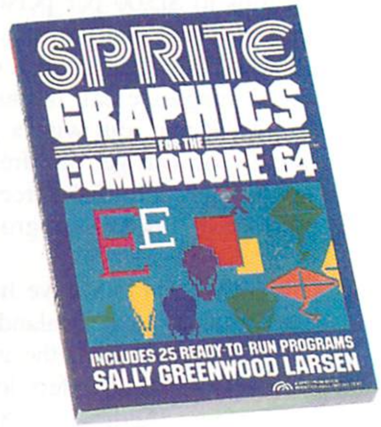
Prentice-Hall speaks a Commodore language other publishers have forgotten. English.*



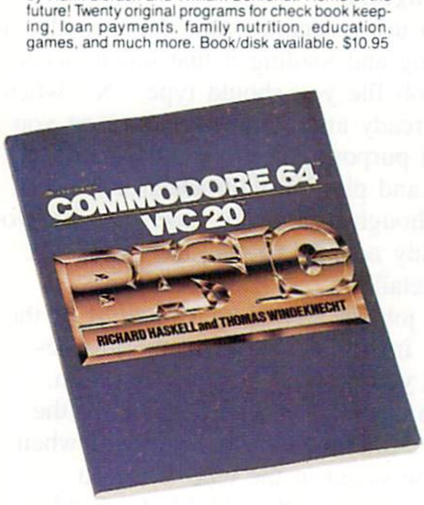
USING THE COMMODORE 64 IN THE HOME by Hank Librach and William Behrendt. Home of the future! Twenty original programs for check book keeping, loan payments, family nutrition, education, games, and much more. Book/disk available. $10.95



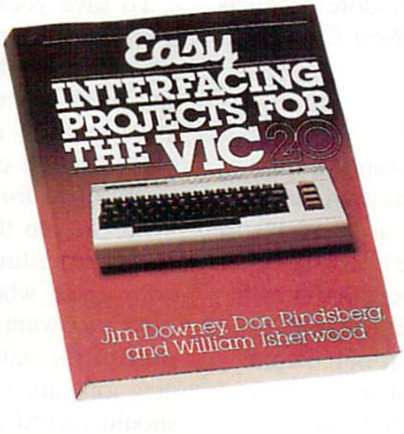
MUSIC AND SOUND FOR THE COMMODORE 64 by Bill L. Behrendt. How to use the Commodore 64's Sound Interface Device and how to write programs that match the sounds of various band instruments. $14.95



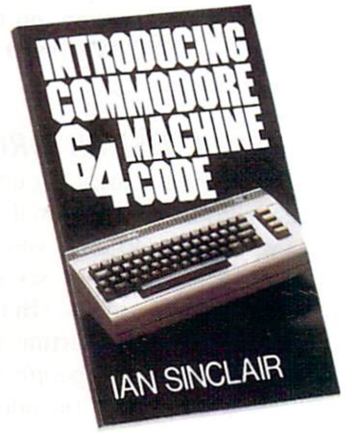
SPRITE GRAPHICS FOR THE COMMODORE 64 by Sally Greenwood Larsen. Shows how to produce high resolution, color, animated graphics. Twenty-five ready-to-run sample programs. $15.95



COMMODORE 64/VIC 20 BASIC by Richard Haskell and Thomas Windeknecht. A top-down programming guide, complete with examples illustrated by video screen photos, that introduces the beginner and advanced user alike to the concepts—and actual writing—of programs in BASIC. $13.95



EASY INTERFACING PROJECTS FOR THE VIC-20 by James Downey, Don Rindsberg, and William Isherwood. Dozens of interfacing projects written in BASIC and specifically designed to maximize the VIC-20's power. $12.95



INTRODUCING COMMODORE 64 MACHINE CODE by Ian Sinclair. This clear, step-by-step intro to programming in machine language also includes sophisticated applications in fast moving graphics and games. $12.95

PRENTICE-HALL BOOKWARE™/THE LEADER IN COMPUTER PUBLISHING

For more information about our computer books and software, write to us at the address below. Dealer inquiries welcome.
Prentice-Hall, General Publishing Division, Englewood Cliffs, N.J. 07632

* We guarantee that all our guides are easy to read and simple to apply without the aid of a reference library, a computer salesperson, or a niece who just graduated from M.I.T.

BUYER'S GUIDE TO VIC AND 64 DATABASES

By Larry Bihlmeyer

Database programs are perhaps the most misunderstood software programs. And they are probably the most powerful and useful.

Perhaps if they were called something else, like electronic file cabinets, they might be less feared. But while "database" and "database management" are terms common to large businesses, they also apply to home and small business use. And that's why we prepared this survey—to help you learn more about what databases do and which ones are available for the Commodore 64 and VIC 20.

First, let's look in simple terms at what a database program is and what it can do. A typical program is made up of records and fields. An easy way to see this is to think of a record as a 3x5 card. If we then draw eight lines on the card to write down information, we have created eight fields on our 3x5 card (one record).

All a database program does, basically, is allow you to create records and fields for each record. Then, by using the speed of a computer, many operations can be made on the stored data, such as sorting and searching for important words or numbers.

Once you've stored your information in your database program (3x5 file card box), you can print out what you've saved in many ways. You can print mailing labels, inventory records with cost information, sport league scores and dates, club membership records, and on and on. The uses for a good database program are endless. And the more you use one, the more uses will develop.

So what databases are available for the VIC and C-64? What do you get for different cost software (and firmware)? Which programs support the RS-232 port?

When you're shopping for a database program, make a list of the uses you need it for. When you've finished your shopping list, compare it with the features included in Table 1. Table 2 provides the manufacturers of these programs.

Database programs basically fall into three price groupings: $15-50, $50-90, $90-150. In general, the more you pay, the more features you'll gain. But as usual, there may be a limit to what you need. Perhaps just a basic database without many frills will be all you need. Just be sure to plan ahead. It's better to buy a program with extra features if you plan to expand your uses or utilize the program for a small business application.

The programs in the $15-50 price range contain the basic database functions. Many are mailing list programs which can also store other data. For example, *Mail Now* ($39.95 for the C-64) allows you to enter seven fields or pieces of information for each record. A typical record would consist of last name, first name, city, state, and zip code. Then, you can have two other fields (Category and Remarks) to store other information such as birthday, customer account number, or golf score. So besides providing a mail list, this program also allows you to keep track of other useful data.

Another low cost database is *Data Manager I* ($24.95) which allows you to create records 30 characters wide by 8 lines deep. Each line then becomes a field. With this format, it is very easy to take your data and put it into a well-organized format for storage and later printing. This particular program uses pointers and indexes to allow you to sort for pieces of information. And if you store numbers, you can also add fields, calculate averages and standard deviations, and do a frequency chart.

A third low cost database is *Info-Manager* ($39.95) which allows you to set up large records (up to 630 characters) and establish up to 18 fields, each with up to 35 characters. The directions with this program are easy to follow, showing you how to set up a record like an index card, with 18 lines maximum, each with 35 characters. Menus help along the way to set up records, make changes, perform disk operations, and print.

Moving up to the next category, the $50-90 group, we find a good example in *Flex File* ($60.00). It consists of a database program, report writer program, and mailing list program: three in one. The database part is very complete, with up to 20 fields

possible and 5 levels of search per field.

You also get complete math functions and logic sorting capabilities. There are menus to help you as you work the program and these direct you to the report writer and mail list. The report writer is like a mini-word processor and allows you to use the data you stored in many different types of reports and letters. The mail list program is primarily a working sample of what you can do with the overall package.

Another program in this second group is *Mini Jini* ($89.95). This comes as a cartridge (firmware) which plugs into the user port. When you turn on your computer, it instantly turns into a database program. You can set up 10 fields for one record, with each field containing up to 75 characters. Other features include alphabetizing a file, searching a file, fixing up a file, doing custom printouts, and disk utilities. And more—it contains a "mathpack" which allows you to add, subtract, multiply, or divide two fields or a field and a constant. This obviously gives you a great deal of flexibility in handling your data.

The third category of programs falls into the $90-$150 price range. *Superbase 64* ($99.00) is a good example. This program is a database management and information retrieval system with a user programmable interface. You can define indexed files of records to suit your needs, recall records quickly with simple or complex search requirements, and process files or sets of records in several different ways.

The number of records on a disk is only limited by the amount of free disk space available. The user can have up to 127 different fields for any one database, with up to 255 characters per field. Maximum record size is 1108 characters, which can cover up to 4 screens of display. Finally, you can have up to

HEADING DEFINITIONS FOR TABLE 1

Tape/Disk/Cartridge	Indicates whether the product comes as a cassette tape, floppy disk, or cartridge.
Record Size (Max)	Shows the maximum size in characters that any one record can be. If a number isn't shown, then the record size is variable.
Number of Fields (Max)	Indicates the maximum size in characters that a field can be. With some programs, the record and field sizes are interrelated, so a variable field size can result.
Field Size (Max)	Indicates the maximum size in characters a field can set up. Sometimes, this is interrelated with the record size and a variable size can result.
Number of Keys	A key is a specific way of labeling a field so it can later be sorted, searched, and operated on. This column will show the maximum number of keys you can name for that program.
Sort Level	Shows the number of levels you can sort and search one specific field.
Math Function	If marked "yes," then the program can perform math functions on fields, such as addition, subtraction, multiplication, and division. Check with each manufacturer for specific details, since not all programs checked "yes" can do all math functions.
Printing	Serial: Printout to the serial port (device #4) is supported. RS-232: Printout to the RS-232 port (device #2) is supported. Std: Only standard formats are available. Cust: Custom printouts can be made.
Logic Sort	Indicates if the program will sort data using logic criteria.
Disk Utility	If marked "yes", the program will allow you to do certain disk utilities while the program is loaded into your computer, such as formatting a disk, getting a directory, and finding a file.
Color	Indicates if the program makes use of the Commodore color capabilities.

TABLE 1: SUMMARY OF VIC 20/COMMODORE 64 DATABASES

Product	VIC	64	Format Tape	Format Disk	Format Cart	Price	Record Size (max)	Number of fields (max)	Field Size (max)	No. of keys	sort level	math fn.	Printing serial	Printing RS-232	Printing std	Printing cust	logic sort	disk util	color	Comments
Inquire Pac		x		x		69.95	200	15	250	VAR	VAR	N	Y	N	Y	Y	Y	Y	Y	
Data Base Manager	 x	x 		x x		89.95 59.95	140	18	24 20	18 9	2 1	Y	Y	N	Y	Y	Y	N	Y	VIC requires 8K expansion
Database Manager		x		x		99.95	2000	200	250	250	1	Y	Y	N	Y	Y	Y	Y	Y	
Data Manager I		x	x	x		24.95	240	8	30	VAR	VAR	Y	Y	N	Y	Y	Y	Y	N	e
Datalog		x		x		39.50	220	12	55	12	VAR	Y	Y	N	Y	Y	Y	Y	Y	
Datman		x		x		59.95	255	19	78	19	19	N	Y	N	Y	Y	Y	Y	N	
Delphi's Oracle		x		x		125.00	8000	99	7000+	99	10	Y	Y	Y	Y	Y	Y	Y	Y	
Filing Assistant		x		x		125.00	256	25	35	10	2	Y	Y	Y	Y	Y	Y	Y	Y	
Flex File		x		x		60.00	254	20	79	20a	5	Y	Y	Y	Y	Y	Y	Y	Y	
Info-Manager		x	x	x		39.95	630	18	35	18	1	N	Y	N	Y	Y	Y	Y	Y	
Inventory	x	x	x			20.00	d	8	d	2	1	N	Y	c	Y	N	N	N	N	
Insta-File b		x		x		61.95 b	1550	62	25	VAR	1	Y	Y	N	Y	Y	Y	Y	Y	Requires Insta-Writer ($51.95)
M'File		x	x	x		95.00	250	32	#	#	#	Y	Y	#	Y	Y	Y	#	#	Tape has fewer features
Magpie		x			x	99.95	VAR	VAR	VAR	VAR	VAR	Y	Y	N	Y	Y	Y	Y	Y	
Mail Now		x		x		39.95	d	7	d	#	1	N	Y	N	Y	Y	N	N	N	Preformatted for mail list data
Mail Pro		x		x		129.95	254	22	99	VAR	VAR	N	Y	Y	Y	Y	Y	Y	Y	Fields preformatted for mail list data
Mini Jini	x 	 x			x x	89.95	350 750	10 10	35 75	0	1	Y	Y	N	Y	Y	Y	Y	Y	
Omnifile	x 	 x	x x	x x		15.95 19.95	VAR	VAR	256	VAR	VAR	Y	Y	Y c	Y	Y	Y	N	Y	Requires 8K for VIC
Powerfile		x		x		89.95	254	20	79	20	5	Y	Y	Y	Y	Y	Y	Y	Y	
Rabbit Base	x		x	•		29.95	vari- able	100	255	1	1	Y	Y	N	Y	Y	Y	N	N	Requires 16K RAM card for VIC
Record Keeper		x		x		99.95	254	80	254	80	20	Y	Y	Y	Y	Y	Y	Y	Y	
Research Assistant	x 	 x	x x	x x		30.00 35.00	VAR	VAR	VAR	12	0	N	Y	Y	Y	N	N	N	Y	Field names preformatted for research data
SDB-64		x		x		39.95	VAR	VAR	VAR	10	1	N	Y	Y	Y	Y	Y	N	N	
Simplifile	x		x			19.95	VAR	VAR	VAR	0	0	N	Y	Yc	Y	N	N	N	Y	
Superbase 64		x		x		99.00	1108	127	255	VAR	VAR	Y	Y	Y	Y	Y	Y	Y	Y	
TOTL. Infomaster		x		x		50.00	2500	10	245	VAR	VAR	Y	Y	Y	Y	Y	Y	Y	Y	
TOTL. Label	x 	 x	x x	x x		20.00 20.00	254	VAR	VAR	0	0	N	Y	Y	Y	Y	N	N	Y	Fields preformatted for label data
Your Filing System	x	x	x	x		49.95	20,- 000	255	80	0	0	N	Y	N	Y	Y	Y	N	Y	Requires 8K VIC expansion

NOTES:
Information not available
VAR Variable value; contact manufacturer for further details.
a Up to 3 recommended for fast sorting
b Requires *Insta-Writer* firmware ($51.95). Also available as part of "Management Combo" package at $99.95.
c "BASIC" program listing can be modified to support printout to Device #2.
d Preformatted record/field sizes—maximum not specified.
e Data Manager II—$49.95. Also available—enhanced program.

15 files per database or one large database. Four menus control program operation in the normal mode. *Superbase 64* also has file ID and password protection as well as two other modes of operation —command mode and programming mode.

Another program in this price range is *Delphi's Oracle* ($125). This program allows you to set up large records (up to 8000 characters) and large fields (up to 99 characters and 9 display pages deep). This obviously is a good program for small business use and has many features common with higher cost business software. It contains the ability to do math functions on the stored data and complex logic sorting. And you can design your own forms on the screen and for printouts.

By now, you can begin to see how the features expand as the cost of the programs increase. When you are shopping, try to narrow your choices down to two or three based on your list of needs. Then contact each manufacturer or local dealer for the lat-

TABLE 2:

MANUFACTURERS LISTING

Product	Company Address
Creat-A-Base	Computer Center 9621 "D" Oates Dr. Sacramento, CA 95827
Inquire Pac	Pacific Coast Software 3220 S. Brea Canyon Rd. Diamond Bar, CA 91765
Data Base Manager	MicroSpec 2905 Ports O'Call Court Plano, TX 75075
Database Manager	Mirage Concepts, Inc. 2519 W. Shaw Suite 106 Fresno, CA 93711
Data Manager I Data Manager II	Timeworks, Inc. 405 Lake Cook Rd. Building A Deerfield, IL 60015
Datalog	Midwest Software Box 214 Farmington, MI 48024
Datman	Software Consultants 4455 Draper St. Salt Lake City, UT 84111
Delphi's Oracle	Batteries Included 186 Queen St. West Toronto, Ontario Canada M5V 1Z1
Filing Assistant	Rainbow Computer Corp. 490 Lancaster Avenue Frazer, PA 19355
Flex File	Webber Software Box 9 Southeastern, PA 19399
Info-Manager	Pyramid Software International 30 Fairfax Suite A San Rafael, CA 94901
Inventory	William Robbins Box 3745 San Rafael, CA 94912
Insta-File (Management Combo)	Cimarron 2158 Hathaway St. Santa Ana, CA 92705
M'File	SJB Distributors 10520 Plano Rd. Suite 206 Dallas, TX 75238
Magpie Simplifile	UMI, Inc. 3503-C Temple Avenue Pomona, CA 91768
Mail Now	Cardco Inc. 313 Mathewson Wichita, KS 67214
Mail Pro	Pro-Line Software 755 The Queensway East, Unit 8 Mississauga, Ontario Canada L4Y 4C5
Mini Jini	Jini Micro Systems, Inc. Box 274 Riverdale, NY 10463
Omnifile	Software To Go 1948 Oak Ridge Turnpike Oak Ridge, TN 37830
Powerfile	City Software 735 West Wisconsin Avenue Milwaukee, WI 53233
Pro-File	A-1 Services 7103 W. Clearwater H-228 Kennewick, WA 99336
RabbitBase	Computer Software Associates 50 Teed Drive Randolph, MA 02368
Record Keeper	Southern Solutions Box P McKinney, TX 75069
Research Assistant Totl. Label Totl. Infomaster	TOTL Software 1555 Third Avenue Walnut Creek, CA 94956
SDB-64	Simpleware 129 Wildbriar Road Rochester, NY 14623
Superbase 64	Precision Software (USA), Inc. Suite 1100 820 Second Avenue New York, NY 10017
Your Filing System	Microtechnic Solutions, Inc. Box 2940 New Haven, CT 06515

est information and to make sure it will work on your system. Double check to make sure it will work with your printer and printer interface.

A good selection will be fun to use and will make you want to find more uses for it. As funny as it sounds, it'll take on its own personality. And that should match yours or you'll be inhibited by it.

So have fun shopping for that database program. It could become the most important piece of software that you own.

RENUMBERING UTILITY

FOR THE C-64

By Anthony Wood

Often while programming, the need arises to insert a line between two lines that have no room between them. In cases like this it would be helpful to renumber the entire program. Other times, when you've finished a program, it would be nice to renumber the program so it looks neat. The renumber utility does this for you.

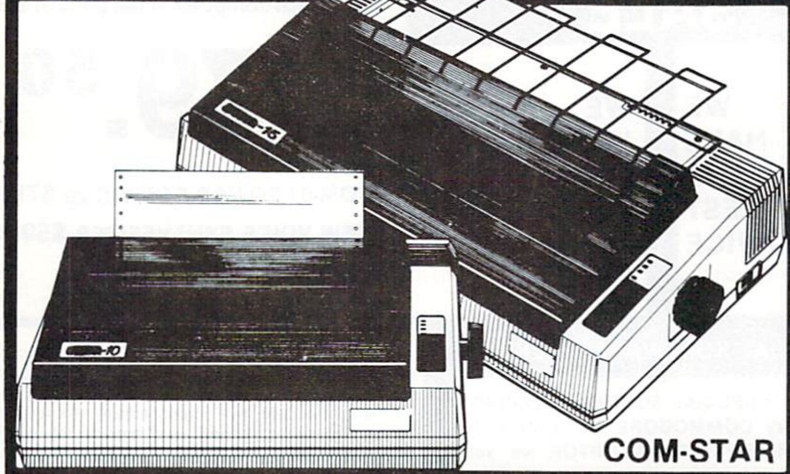
The renumber program will renumber an entire program while changing GOTO, IF-THEN, GOSUB, ON GOTO, and ON GOSUB references to the correct values.

When you run the program, you are asked for the name of the program to renumber. Enter the name you saved your program under. Next, you are asked for the name of the renumbered program. As the renumber utility renumbers your program, it will be saved under this file name. Next, you are asked what you want the first line in the renumbered program to be. After you enter this, you are prompted for the increment between lines. Once you have answered these questions, the renumber utility proceeds to renumber your program.

SEE PROGRAM LISTING ON PAGE 93

(See Next 0 Pages)

80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor Friction Printer

only $ 179**

•15 Day Free Trial -180 Day Immediate Replacement Warranty

- **Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.**
- **Fast 80-120-160 Characters Per Second**
- **40, 46, 66, 80, 96, 132 Characters Per Line Spacing**
- **Word Processing**
- **Print Labels, Letters, Graphs and Tables**
- **List Your Programs**
- **Print Out Data from Modem Services**
- **"The Most Important Accessory for Your Computer"**

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal paper! Best thermal printer in the U.S.A.! (Centronics Parallel Interface).

**DELUXE COMSTAR T/F 80 CPS PRINTER—$179.00

The COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interface).

Premium Quality—120 CPS COMSTAR T/F SUPER-10X PRINTER—$289.00

COMSTAR T/F (Tractor Friction) SUPER-10X PRINTER gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters, plus 2K of user definable characters! The COMSTAR T/F SUPER-10X PRINTER was Rated No. 1 by "Popular Science Magazine." It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX 80).

Premium Quality—120 CPS COMSTAR T/F SUPER-15½" PRINTER—$379.00

COMSTAR T/F SUPER 15½" PRINTER has all the features of the COMSTAR T/F SUPER-10X PRINTER plus a 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100).

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 10" PRINTER—$399.00

SUPER HIGH SPEED COMSTAR T/F (Tractor Friction) PRINTER has all the features of the COMSTAR SUPER-10X PRINTER plus SUPER HIGH SPEED PRINTING—160 CPS, 100% duty cycle, 8K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs. RED HOT BUSINESS PRINTER at an unbelievable low price!! (Serial or Centronics Parallel Interface)

Superior Quality SUPER HIGH SPEED—160 CPS COMSTAR T/F 15½" PRINTER—$529.00

SUPER HIGH SPEED COMSTAR T/F 15½" PRINTER has all the features of the SUPER HIGH SPEED COMSTAR T/F 10" PRINTER plus a 15½" carriage and more powerful electronics to handle larger ledger business forms! Exclusive bottom paper feed!!

PARALLEL INTERFACES

For VIC-20 and COM-64— $69.00
For Apple Computers— $79.00

NOTE: Other printer interfaces are available at computer stores!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

Add $17.50 for shipping, handling and insurance. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Reader Service No. 15

4 Color

80 COLUMN Letter Quality PRINTER / PLOTTER

Super Special

$99

1/2 PRICE SALE



$99

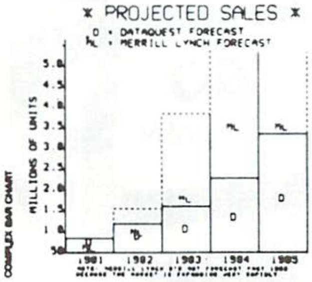
1/2 PRICE SALE

• LOWEST PRICE IN U.S.A.

Commodore-64 & VIC-20

· $500 Software SAVINGS Coupon Included ·

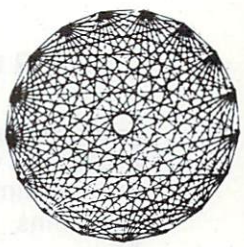
• List your programs • High resolution graphics for bar charts and geometric figures (like spirograph) • Plugs directly into VIC 20 and Commodore 64 — Interface included • Lowest cost letter quality printer in the country.



ACTUAL PRINT SAMPLES

UPPER CASE – ABCDEFGHIJKLMNOPQRSTUVWXYZ

LOWER CASE – abcdefghijklmnopqrstuvwxyz



At last you can list your programs (even control characters) and make beautiful high resolution graphics at an affordable price. This **80 column letter quality printer/plotter** is great for making complex bar charts for business plus fancy greeting cards and geometric designs. Great for homework too. **Everyone must have a 4 color printer plotter for their VIC-20 or Commodore-64.** List $199.00. **Sale $99.00.** · $500 Software SAVINGS Coupon Included ·

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa · MasterCard · C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COMMODORE-64 or VIC-20

VOICE SYNTHESIZER

NEW NEW

MAKE YOUR COMPUTER TALK

VOTRAX BASED HARDWARE

SALE ONLY **$59.00**

You can program any words or sentences • Adjust volume and pitch • Make adventure games that talk • Real sound action games • Make customized talkies • (Demo disk or tape included)

You can add **TEXT TO SPEECH SOFTWARE** that allows you to simply type what you want to hear!! Also allows you to add sound and voice to **SCOTT ADAMS AARDVARK** and **"ZORK" ADVENTURE GAMES** List $29.95 Sale $19.95 (Disk or Tape).

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

COMMODORE 64

80 COLUMN BOARD

SALE

FOR ONLY **$99.00**



Now you can program 80 columns on the screen at one time! Converts your Commodore 64 to 80 columns when you plug in the PROTECTO 80 Expansion Board. List $199.00. **Sale $99.00.**

FOR ONLY $24.95 you can get an 80 Column Board "WORD PROCESSOR" with mail merge and terminal emulator PLUS! AN ELECTRONIC SPREAD SHEET (like Visicalc) List $59.00. **Sale $39.90.**
*If purchased with board only $24.95. (Tape or Disk.)

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master Card. We ship C.O.D.

WE LOVE OUR CUSTOMERS

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

EXPAND YOUR VIC-20 TO COMMODORE-64 POWER!

FOR ONLY $79⁰⁰

(PLUS A FREE $29.95 CARTRIDGE GAME) LIST PRICE $159.95

Just plug in the **32K RAM MEMORY EXPANDER** and you have as much usable programming memory as the Commodore-64 computer!!! This **32K RAM EXPANDER CARTRIDGE** gives your VIC-20 computer **9 TIMES MORE PROGRAMMING POWER!!**

5 YEAR WARRANTY



FEATURES

- *Allows Full 32K Memory Expandsion*
- *Separate Enable Switch for each 8K Block*
- *Gold Plated Contacts*
- *Premium Grade Components*
- *Fully Tested and Burned In*
- *5 Year Warranty*
- *Complete Documentation*

1/2 PRICE EXPANSION SALE!!

	LIST	SALE
3K Ram Expander	$ 39.95	$19.95
8K Ram Expander (2½ times more power)	$ 69.95	$34.95
16K Ram Expander (4 times more power)	$ 99.95	$49.00
32K Ram Expander (9 times more power) (PLUS $29.95 FREE GAME CARTRIDGE)	$159.95	$79.00
3 Slot Expander Board — Switch Selectable	$ 49.95	$24.95

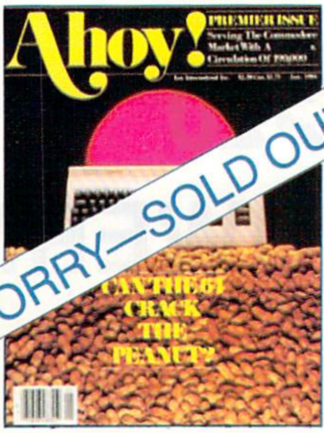
WE LOVE OUR CUSTOMERS!

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

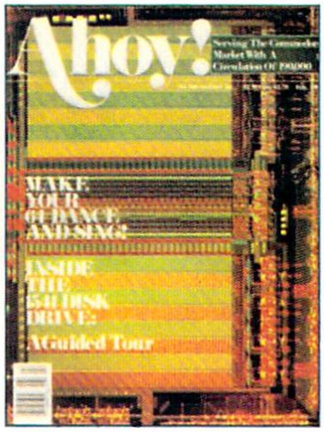
Ahoy!

Back Issues



ISSUE #1—JAN. '84 $4.00
The 64 v. the Peanut! The computer as communications device! Protecto's Bill Badger interviewed! And ready to enter: the Multi Draw 64 graphics system! The Interrupt Music Maker/Editor! A Peek at Memory! Programming Sequential Files!

Don't punch another key without a complete collection of *Ahoy!* and the programming strategies and product analyses each issue provides. Order while supplies last!



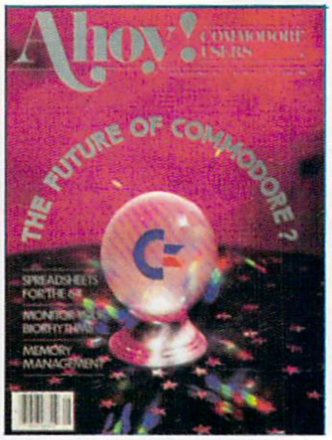
ISSUE #2—FEB. '84 $4.00
Illustrated tour of the 1541 disk drive! Synapse's Ihor Wolosenko interviewed! Users groups! Artificial intelligence! And ready to enter: Music Maker Part II! Night Attack! Programming Relative Files! Screen Manipulation on the Commodore 64!



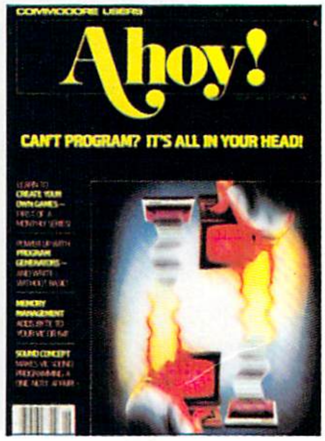
ISSUE #3—MAR. '84 $4.00
Anatomy of the 64! Printer Interfacing for the 64 and VIC! Educational software: first of a series! Commodares! And ready to enter: Space Lanes! Random Files on the 64! Easy Access Address Book! Dynamic Power for your 64!



ISSUE #4—APR. '84 $4.00
Petspeed and Easy Script tutorials! Printer interfacing and educational software guide continued! Lower case descenders on your 1525! Laserdisc! The Dallas Quest Adventure Game! And ready to enter: Apple Pie! Lunar Lander! Name that Star!



ISSUE #5—MAY '84 $4.00
The Future of Commodore! Inside BASIC program storage! C-64 Spreadsheets! Memory Management on the VIC and 64! Educational Software Guide continues! And ready to enter: Math Master! Air Assault! Biorhythms! VIC 20 Calculator!



ISSUE #6—JUNE '84 $4.00
Game programming column begins! Program generators! Rupert on inputting! Memory Management and Educational Software Guide continue! And ready to enter: Post Time for the 64 and 20! Alpiner! Sound Concept for the VIC 20!

Send coupon or facsimile to:
Ahoy! Back Issues, Ion International Inc., 45 West 34th Street—Suite 407, New York, NY 10001

Ahoy!

Please Send Me The Following:
_____ Copies of issue number _____
_____ Copies of issue number _____
_____ Copies of issue number _____

Enclosed Please Find My Check or Money Order for $__________

(Outside the USA please add $1.00 for every copy)

NAME__________
ADDRESS__________
____
CITY__________
STATE__________
ZIP CODE__________

TRENNING YOUR CURSOR
TRAINING YOUR CURSAR
TRAINING YORE CURSOR
TRAINING YOUR CURSOR

By Michael Kleinert and David Barron

This month we thought we would take a step back into the past to a time when we first began using Commodore computers. That first 8K PET with the calculator keyboard possessed four puzzling keys above the numeric keypad. It didn't take us long to figure out that these keys were used for "onscreen" editing of your program. It wasn't until later that the real power of these keys was unleashed. It just so happens that Commodore has maintained the function of these keys in all of their computers, including your VIC or 64.

These keys are used to control cursor movement and positioning from within a program. Even when a program is running and you do not actually see a cursor on the screen, the computer is keeping track of where the "cursor" should be. This is where the next printed item will appear.

When a print statement is executed, with each character displayed, the cursor is moved over one character to the right. When the computer is finished with the line, the cursor moves back to the left hand column, and down to the next line. If you happen to tag a semicolon to the end of the print statement, the cursor stays at the position following the last printed character.

```
10 PRINT "HELLO,"
20 PRINT "THIS IS A SAMPLE RUN WI
THOUT A SEMI-COLON."
30 END

RUN

HELLO,
THIS IS A SAMPLE RUN WITHOUT A SE
MI-COLON.

10 PRINT "HELLO, "
20 PRINT "THIS IS A SAMPLE RUN WI
TH A SEMI-COLON."
30 END

RUN

HELLO, THIS IS A SAMPLE RUN WITH A
SEMI-COLON.
```

If you have ever tried typing any of the cursor control keys after typing quotation marks, you have probably found that strange symbols appear. When you type an open quotation mark, the built-in editor goes into "quotes mode." This means that when you now type any of the cursor controls, including CLEAR SCREEN and HOME, it will not respond with its usual function, but will instead print its identifying symbol. (Function keys, color keys, INSERT, RVS ON and RVS OFF produce their own symbols as well.) Below is a chart of these symbols.

CURSOR UP	**CURSOR RIGHT**
CURSOR DOWN	**HOME CURSOR**
CURSOR LEFT	**CLEAR SCREEN**

The up, down, left, and right cursor functions move the cursor one space in their respective direction. This is a "non-destructive" cursor movement. In other words, the cursor will be moved without erasing anything in its path. This enables you to move the cursor anywhere on the screen without any ill effects. The clear screen control clears the screen and returns the cursor to the upper left corner. Home puts the cursor in the upper left corner without clearing the screen.

You may be wondering how you can use these cursor control functions to your advantage. They can be used to format text on the screen neatly, or to update data which is being displayed on the screen without erasing the screen and reprinting everything.

Below is a sample program that displays the constantly changing built-in clock and a running counter simultaneously using cursor controls.

```
10 PRINT"{SC}"
20 PRINT"{HM}{CD}{CD}{CD}{CD}{CR}
{CR}{CR}{CR}{CR}";TI$;"{CR}{CR}{C
R}{CR}{CR}";C
30 C=C+1
40 GOTO20
```

Without knowledge of cursor controls, to clear the screen and print an asterisk in the 10th column of the 5th row of the screen, you might be inclined to write the following program.

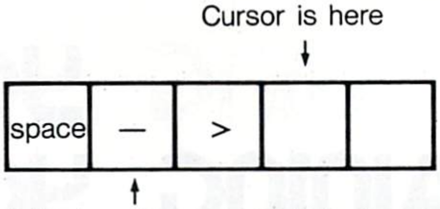
```
5 PRINT"{SC}";
10 FORX=1TO5
20 PRINT
30 NEXT X
40 PRINT TAB(10);"*"
50 END
```

However, with cursor controls, the above program can be accomplished with just one print statement as shown below.

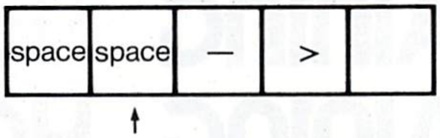
```
10 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{
CR}{CR}*"
20 END
```

In the print statement above, the first character clears the screen and moves the cursor to the upper left hand corner. The next five characters move the cursor down five lines, and the next ten characters bring the cursor over ten spaces to the right. The asterisk is then printed in this position.

The cursor control characters can also be used to create some simple animation. Let's say we want to move the two characters "—>" across the screen. The technique for moving this "arrow" is fairly simple. It must be printed, then the tail must be erased, and it is then printed again over one screen position. A trick used in doing this is to always print a space before the arrow. Using this trick, in moving over one screen position and reprinting the arrow, the space will always erase the old tail. This is a bit hard to follow at first, so below is a diagram to help clarify this.



Print two "cursor lefts" to move cursor here



Printing a space and then the arrow erases the old tail and "moves" the arrow to the right

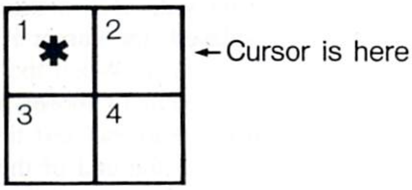
The following program imitates the above steps to move the arrow:

```
10 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}
{CD}";
20 FOR X = 1 TO 35
30 PRINT" ->{CL}{CL}";
40 FOR Y = 1 TO 50 : NEXT Y : REM
 DELAY LOOP
50 NEXT X
60 END
```

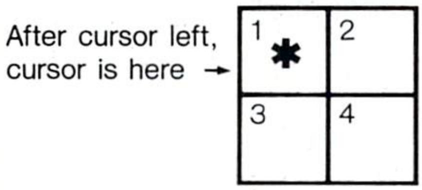
Moving a character in a vertical direction is a little bit different. To move an asterisk down the screen, we must repeat the following steps:

1. Print the asterisk
2. Erase the asterisk
3. Move the cursor down underneath the previous asterisk
4. Go back to step 1

Imagine that the following diagram represents four positions on the screen.



After printing the asterisk in box 1, the cursor will be positioned in box 2. Therefore, to erase the asterisk we must move the cursor back to box 1 with a cursor left.



Continued on page 48

***Next issue, David Barron delves into one of your Commodore 64's most amazing facets: its sound-production capabilities. His two-part article/program will be your key to* Unraveling the Mysteries of Sound!**

COMMODARES

Programming Challenges



by Dale Rupert

Each month, we'll present several programs designed to toggle the bits in your cerebral random access memory. We invite you to send your solutions to Commodares, P.O. Box 723, Bethel, CT 06801. We'll print and discuss the cleverest, simplest, shortest, or most unusual solutions. Your original programming problems would be equally welcome!

Problem #7-1: Definite Ending

This problem was submitted by Paul T. Dawson of Springhouse, Pennsylvania. Paul's problem is to write a one-line program which does not GOTO itself or RUN itself. It must create a changing display. And (here's the catch) it must take as long as possible to run! It must eventually end on its own, but that "eventually" may be long after you and I are gone.

Paul sent his solution which we will list next month. He claims his one-liner will take 2E+134 years to finish. That's a good start, but surely some other *Ahoy!* readers can put that number to shame. Your solutions should also include an explanation for how the duration was calculated. (Deadlines won't allow us to run them all to completion!)

Problem #7-2: Unscrambled Word

This problem was submitted by Kent Rittenhouse of Wauseon, Ohio. Kent's problem is to write a program for which you can enter a scrambled word, and the computer will print out all possible combinations (or permutations, to use the technical term) of the letters of that word.

We won't require that the computer be able to decide which of the combinations is the correct, unscrambled word. You would easily be able to select the correct word from among the 3.6 million or so (10 factorial to be exact) combinations resulting from any ten-letter word you might enter. DOGO KULC!

Problem #7-3: Letter Count

The user types a sentence into the computer, and the computer responds with the number of letters in the sentence. Punctuation marks and spaces are not counted.

Problem #7-4: Word Count

The user types a sentence into the computer, and the computer responds with the number of words in the sentence. Easier said than done?

Because of time delays between the time this column is written and the time it is actually published (this being written on a chilly March evening), it will be a while before there are any solutions submitted for last month's *Commodares*. Consequently we will present the results of *Problem #6-1: Speed Demon* once we have some results.

Problem #6-2: Simple Mystery probably didn't fool too many of you. It merely allowed the user to type, while any vowels that were typed appeared in reverse video on the screen.

A solution to *Problem #6-3: String Challenge* is listed below. The subroutine at line 100 determines the starting location, if any, of the string A$ within the string B$. It assigns the value of that starting position to N. Line 150 is executed only if A$ is found within B$. QQ then stores the value we want for N. In order to terminate the FOR-NEXT loop properly, QQ is assigned the maximum value for the loop. That way when the NEXT statement is executed, the program continues with line 170. This is a way of cleaning up after yourself when you leave a FOR-NEXT loop prematurely.

```
10 REM > PROBLEM #6-3 : STRING CH
ALLENGE
20 A$="CAT":B$="CONCATENATE"
30 GOSUB 100
```

```
40 PRINT N
99 END
100 ::REM > INSTRING SUBROUTINE
<
110 LA=LEN(A$):LB=LEN(B$)
120 N=0
130 FOR QQ=1 TO LB-LA+1
140 IF A$ < > MID$(B$,QQ,LA) THEN
 160
150 N=QQ : QQ=LB-LA+1
160 NEXT QQ
170 RETURN
```

There probably won't be any one-liners to do *Problem #6-4: Graphic Rectangle.* The program below first assigns to variables the various graphic symbols that will be used. Line 80 puts the correct space at the top of the screen. It divides the total unused vertical space (24-H) in half. Line 110 determines the amount of open space to the left (LS). Line 120 puts the space there. Line 130 prints the Upper Left corner. Lines 140-160 print the Top. Line 170 prints the Upper Right corner. The sides are printed by lines 180-200. Lines 210-260 put on the bottom.

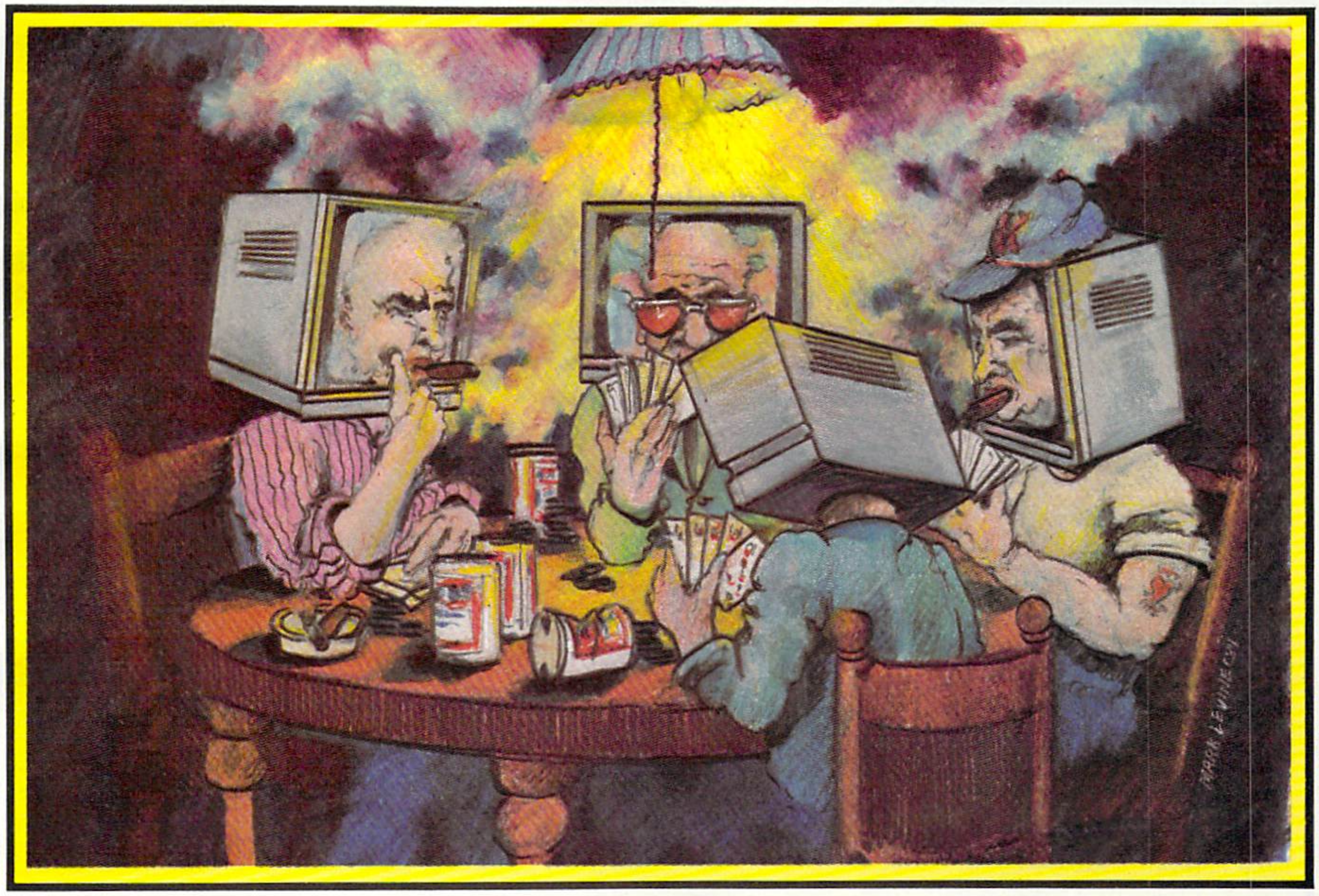
```
10 REM PROBLEM #6-4 : GRAPHIC REC
TANGLE
20 UL$=CHR$(111):UR$=CHR$(112)
30 LL$=CHR$(108):LR$=CHR$(186)
40 T$=CHR$(183):B$=CHR$(175)
50 L$=CHR$(165):R$=CHR$(167)
60 INPUT "HEIGHT, WIDTH";H,W
70 PRINT CHR$(147)
80 FOR N=1 TO (24-H)/2
90 PRINT
100 NEXT
110 LS=(40-W)/2
120 PRINT SPC(LS);
130 PRINT UL$;
140 FOR N=2 TO W-1
150 PRINT T$;
160 NEXT
170 PRINT UR$
180 FOR N=2 TO H-1
190 PRINT SPC(LS);L$;SPC(W-2);R$
200 NEXT
210 PRINT SPC(LS);
220 PRINT LL$;
230 FOR N=2 TO W-1
240 PRINT B$;
250 NEXT
260 PRINT LR$
```

We are still receiving solutions to the *Palindromes* and *Reversals* problems from the March *Ahoy!* Paul Dawson combined the *Crossed Name* and *Reversals* to come up with the following program.

```
10 REM > CROSSED AND REVERSED NAM
E <
20 INPUT A$:PRINT CHR$(147):L=LEN
(A$)
30 FOR A=1 TO 9E9 : FOR B=0 TO 38
-L
40 FOR C=1 TO L
50 PRINT CHR$(145)TAB(B+C)MID$(A$
,C,1)
60 PRINT CHR$(145)TAB(39-B-C)MID$
(A$,C,1)
70 NEXT : PRINT : NEXT : NEXT
```

Thanks to the following people for their work on *Palindromes* and/ or *Reversals:* Ray Winter (Savage, MT), Michael Vail (Thompson, PA), Barbara Steinman (New York, NY), A.L. Russ (Auburn, KS), R. Shane Robinson (Avoca, IA), Kent Rittenhouse (Wauseon, OH), Dave Playford (Brockville, Ontario), Charles Morgan (Los Angeles, CA), Steve Molloy (Chelmsford, MA), Richard Magoun (Plympton, MA), Wilf Long (Manitowaning, Ontario), John Eveland (Des Moines, IA), Rob Einer (Woodruff, WI), Paul Dawson (Springhouse, PA), Tony Damato (Jacksonville, FL), Joe Cook (Paw Paw, MI), Robin Boyd (Lakewood, NJ), Robert Bostwick (No. Platte, NE), and John Adams (St. Charles, IL).

Their solutions varied from long to short, user-friendly to bare-necessities, and with differing degrees of flexibility. Thanks for all the kind words and suggestions, too. Till next month, keep those brains working! □


© Mark Levine 1984

BRISK

FOR THE C-64

By Joseph Fusco

Most card games depend primarily on luck. *Brisk* is one in which you must rely on strategy, luck, and memory. You must try to outwit the computer, be lucky with dealt cards, and remember cards played and points accumulated. The game could be programmed for two players, but I felt computers make very few mistakes and present a formidable opponent.

Type in the program, then save a copy to disk or tape before running in case of typing errors! Typing errors may cause your program to crash, losing hours of typing.

The following will give you an idea of how to play the game.

The object of the game is to be the first player to score 121 points. The game is played in two hands, each consisting of twenty rounds and worth 120 points. You must try to score as many points as you can in each hand. The more points you make in the first hand, the less you will have to make in the second hand. As soon as one of the players reaches 121 points the game is over! Playing the game well takes a little practice. After reading the instructions you will have the fundamentals of how to play.

The game is played with forty cards. There are no 8's, 9's, or 10's in the playing deck.

Point values for each card are as follows:

1) ACES are worth eleven points.
2) THREES are worth ten points.
3) KINGS are worth four points.
4) JACKS are worth three points.
5) QUEENS are worth two points.

All other cards have no point value. In this card game the JACKS are higher than the QUEENS.

The computer deals seven cards at the start of each hand: three to you, three to itself, and one face up.

Trump is determined by the card that is face up

Continued on page 48



AN INDEPTH REPORT

By Morton Kevelson

While dual disk drives are nothing new to the veterans of microcomputing, many of our readers are still novices in this regard. A good percentage are probably still considering the aquisition of their first disk drive. Just what is a dual disk drive and how does it differ from the Commodore 1541? What does it do to justify the premium price tag, more than twice that of two 1541 disk drives? Is it twice as hard to figure out as the 1541? One would certainly hope not, as figuring out the 1541 was bad enough.

Read on. The answers to these questions and more will be forthcoming.

WHY DUAL DISK?

The Commodore 1541 disk drive is a very powerful data storage device, capable of storing over 170,000 bytes of data on the surface of a magnetically coated floppy disk. Even more important, any of this data can be rapidly and randomly accessed by the computer. This large storage capacity, combined with direct access to data, easily justifies the purchase of a disk drive.

This being the case, what advantage does a second disk drive offer? The most obvious is a doubling of accessible storage capacity. The widespread acceptance of the Commodore 64 has resulted in a concurrent development of more sophisticated software. These programs are very often too complex to reside in the computer's memory and still leave space for data. Software developers are able to work around this limitation by breaking up the programs into functional modules which are stored on the disk and called when needed. (See *Memory Management Parts I & II,* May and June *Ahoy!,* for

examples of these techniques.)

In many cases, commercial programs are copy protected, making it inadvisable or impossible to store data on the same disk. Other commercial packages are so complex that little or no space is available on the program disk for data storage. In either case, the user must swap disks each time a data or program operation is called for: a tedious procedure at best, aggravatingly time-consuming at worst. Having two drives, one for program and one for data, is a clear advantage.

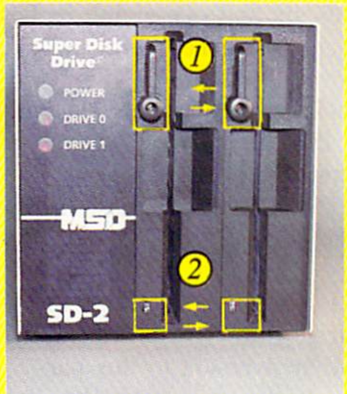


PHOTO 1: Front panel of MSD Dual Disk Drive. Disks are inserted vertically and secured by twisting and locking levers a quarter turn. READER SERVICE NO. 4

Shortly after acquiring a disk drive, the need for maintaining backup copies of both program and data files becomes obvious. With regard to program files, a simple LOAD, swap disk and SAVE will usually work. The process becomes more involved with non-relocatable machine language programs and various types of data files. The latter always require some kind of custom program just to be read. These copy programs always require the use of the computer for their operation. For single drive owners these utilities are essential.

Although the solution to backing up most disks is readily available, it is not particularly convenient. The process tends to be time consuming. The primary limitation is the speed at which data can be transferred between the computer and the disk drive. In the case of the Commodore 64 and the VIC 20, the connecting link is the serial bus. To lower production costs, Commodore designed this link with a minimum number of wires. Data is transferred over a single conductor one bit at time. Compare this to the eight wires, which transfer a full byte at one time, inside the computer. To enhance the reliability of the link, and keep the price down, the transmission speed was kept slow. As a result, LOAD and SAVE operations do take time.

Slow operating speeds are fairly easy to live with when loading and running programs. Backing up data is another story. This process, although essential, is basically nonproductive. Typical backup times for an entire disk are on the order of twenty to forty minutes. The time varies depending on the particular software being used. Additionally, during the backup procedure, the computer is tied up. It must act as both a controller and an intermediate storage area for the data being duplicated.

A single disk drive poses an additional bother. The computer has considerably less memory than the capacity of a disk. The user must attend the entire procedure, swapping disks four or more times. Two single disk drives offer some relief. The user is able to "walk" away and let the computer do all the work. Two single drives do not speed things up very much because the data must still pass through the computer via the same serial bus. The computer again remains tied up for the duration.

Consideration must also be given to the fact that the useful life of the disk drive is being frittered away while duplicating disks. Although the 1541 appears to be designed for continuous operation, long copy sessions will raise the operating temperatures

PHOTO 2: Back of the MSD has all the standard 1541 connectors, including two serial port connectors that allow daisy chaining of additional disk drives.



to their maximum. Rumor has it that many users have found the drive's limitations the hard way.

THE MSD SD-2

The MSD SD-2 is a true dual disk drive. This means that the two drive mechanisms share a common controller and have available a common internal bus for high speed communication. The advantage of the internal bus is that it does not utilize the computer as an external data link. After all, computers have lots of better things to do than act as a chit-chat link for a couple of disk drives. Furthermore, having a common controller allows the disk

operating system to have the necessary copy commands built right in. Once again, the computer can go about its business without bothering with all those copy programs.

The MSD SD-2 has these capabilities. In addition to all the 1541 DOS commands, the Duplicate command is also supported. This command is part of the Commodore DOS on the now discontinued CBM 4040 dual disk drive, as well as all the other Commodore dual disk drives. Executing the command will cause the disk drive to make an exact

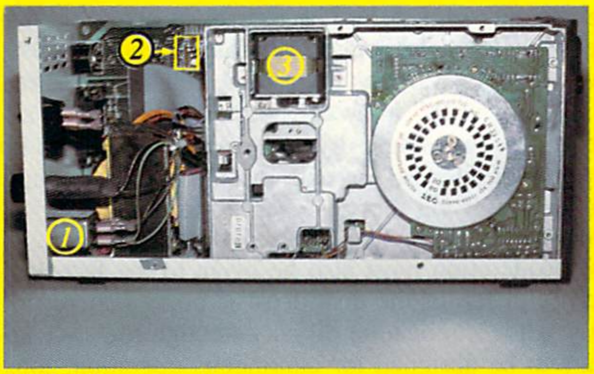


PHOTO 3: removing cover reveals two floppy units.

duplicate of the original disk. The syntax is straightforward:

```
OPEN 15,8,15,"D1=0"
```

Or if the command channel is already open:

```
PRINT#15,"D1=0"
```

If the DOS wedge has been initialized,

```
/D1=0
```

does the job. In each case the disk in drive 0 will be duplicated onto the disk in drive 1. Note the use of drive numbers 0 and 1. The duplicate command works in either direction by simply reversing the drive numbers in the command.

Both drive units are still addressed with a single device number. The individual drive units are selected by the drive number parameter (dr) embedded in the actual DOS command. Users of the 1541 single disk drive may have noticed references to this parameter in their manual without paying too much attention to it. The parameter is rather superflous with a single drive.

The duplication process is farily quick. The total time involved, including formatting of the destination disk, is one minute and fifty four seconds!

Yes, that is less than two minutes. When you consider that a 1541 takes a minute and a half just to format a disk, this is quite an improvement.

Hold on now! Don't go off in search of your checkbook, credit cards, or local bookie. This is just the icing on the cake. Before we get into the rest of the operating details, let's take a break and look at the hardware.

THE HARDWARE

The MSD SD-2 is a boxy looking affair in a sturdy steel case. This case also acts as a heat sink for the drive electronics, keeping things from getting too hot. The drive stands six and one-eighth inches high by five and seven-eighths inches wide and thirteen inches deep. The overall volume of 468 cubic inches is just 15 percent larger than the 407 cubic inches occupied by the 1541 disk drive.

The front panel is shown in photo 1. The two drive units are vertically oriented. The disks are inserted vertically with the labels to the left. As shipped from the factory, drive 0 is to the left and drive 1 is to the right. If you are a veteran CBM 4040 user, which has drive 0 to the right, or are left handed, or just have an aversion to this particular arrangement, the drive numbers can be easily changed. We shall see how shortly. For some reason, MSD chose not to supply any drive number labels for the front panel—a minor oversight.

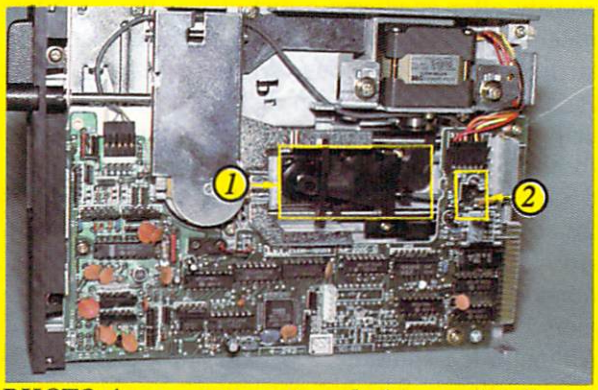


PHOTO 4: connectors set each drive unit to 0 or 1.

The disks are secured by the positive action of twisting the locking levers (item 1, photo 1) a quarter turn clockwise. There are no spring loaded doors or disk ejection mechanisms with this arrangement. The disk openings are properly recessed to permit a firm finger grip on the disk jacket. This allows the user to overcome any adverse effects due to loose labels or write protect tabs.

You will also notice, in photo 1, the presence of five indicator lights. Some of these light-emitting

diodes (LED) are the indicators that Commodore left out on the 1541. The green Power LED, in the upper left hand corner, serves a dual purpose. A built-in diagnostic routine will identify certain component failures by cyclically blinking a number of times on power up. The number of flashes in each cycle (1 to 7) identifies the particular failed component. These could be the microprocessor, the RAM or ROM chips, or the drive mechanism. This feature could save a service call or shipment back to the factory for a knowledgeable user. During normal operation this light glows continuously.

The two drive LEDs, labeled Drive 0 and Drive 1, indicate when a disk file is open on the respective drive unit. This can be very useful for a programmer when debugging is being done. These lights will also blink when a disk error condition occurs. These lights actually perform the same function as the red LED on the 1541 disk drive.

In addition, each drive unit has a red activity LED (item 2, photo 1). These indicate which unit is physically active at any given time. They will light up whenever the drive motor for the corresponding unit is spinning.

The back of the disk drive (photo 2) has all of the standard 1541 connectors. The two serial port connectors (item 1, photo 2) allow the daisy chaining of additional disk drives and a printer. The power connector (item 2, photo 2) is identical to the one on the 1541 as well. And as with the 1541, the power switch is inconveniently located on the back panel. The most important difference back here is the presence of the 24 pin "D" connector (item 3, photo 2). This connector is a fully implemented 8-bit parallel IEEE-488 interface. The presence of this interface makes the MSD SD-2 fully compatible with all of the Commodore PET and CBM series computers, as well as the Commodore 64 and VIC 20. Therefore no additional adaptors are required to use the disk drive with any of Commodore's current line of computers. The IEEE-488 interface, being a parallel interface, allows significantly faster data transfers than the serial bus—two to three times faster as it turns out. More on this later.

(For the curious in our audience, IEEE stands for The Institute of Electrical and Electronic Engineers. This is one of the privately run, non-profit national engineering organizations. Among other things, the IEEE has assumed the responsibility of publishing standards for use as voluntary guides to promote uniformity in industry. The term IEEE-488 refers to one of these standards.)

Maintenance of the disk drive should be a serviceperson's dream. The steel cover can be easily removed by extracting the three screws on each side of the case. Doing so reveals the two half-height floppy units by T.E.C. (Tokyo Electric Works) (photo 3) and a nicely laid out main circuit board. These components, along with the power transformer on the main chassis, make up the entire works of the disk drive.

Attention to detail and reliability is evident. The incoming power line is fully suppressed by a self-contained power line filter (item 1, photo 3). Due to this, no external surge protection is required with the disk drive. The drive's device number can easily be changed by simply removing one or both of the convenient jumper plugs on the main circuit board (item 2, photo 3). There are no wires or circuit traces to cut here.

The large disk, with the radial strobe markings, is a quartz-locked, brushless dc disk motor, which spins the disk at a very constant 300 revolutions per minute. The read/write head stepper motor (item 3, photo 3), which positions the read/write head to the required track, is slightly above and to the rear. Each drive module has its own control electronics, which simplifies the tasks performed by the main control board.

As with all Commodore peripherals, the MSD SD-2 is an "intelligent" drive. The DOS is completely contained in 16 kilobytes of ROM on the main circuit board. The heart of the SD-2 control system is a 6511Q microprocessor. DOS work space and buffers are provided for by 6 kilobytes of onboard RAM.

The individual drive units can easily be removed (photo 4) for servicing by simply extracting two screws from the bottom of the case. Two cable connectors at the back of the unit complete the job. Once removed, head cleaning can easily be done as described in *Inside the 1541 Disk Drive: A Guided Tour* (February *Ahoy!*). The pressure pad arm assembly is item 1, in photo 4.

At this point the drive numbers can be changed as mentioned earlier. Each drive unit has a push on connector (item 2, photo 4) which sets it to drive 0 or to drive 1. By simply resetting these connectors, the unit can be configured with drive 0 to the right. The drives should be left in their original physical position, since the order in which they are connected is important.

The overall operation of the drive is slightly noisier than the 1541. One noise, however, is conspicuous by its absence. The drive never makes the

loud chattering sound, associated with the 1541, whenever a disk is formatted or a disk error is detected. This is because head position is optically sensed by the MSD drive. It is not necessary to knock the head positioning assembly against a fixed stop to determine its location. This feature should significantly lengthen the interval between head alignments.

MORE ON THE DOS

The MSD SD-2 DOS is 100 percent compatible with all Commodore disk operations and formats associated with the CBM 4040, 2031, 1540, and 1541 disk drives. All Commodore DOS commands are fully implemented. The only problems we encountered dealt with software which made direct calls to routines within the DOS itself. For example, the copy protection on the Electronics Arts Software series seems to use direct calls, as these disks would not load on the MSD. This is really not a significant drawback. There is no guarantee that future versions of the Commodore 1541 ROM's will maintain this compatibility either.

More important to compatibility is the availability of the serial bus. Absolutely no additional hardware, or software, is required to access all of the dual drive capabilities with a VIC 20 or a Commodore 64. Thus all of the computer's memory is available for application programs.

The SD-2 really shines when used with software which recognizes a dual drive, such as *Calc Result Advanced.* The first time we used this program, it took nearly one hour to initialize the program disk and create a backup copy. With the SD-2, the entire process took less than five minutes!

The Commodore CP/M card works very well without the need for an IEEE-488 adaptor. Simply plug in the CP/M card and the SD-2. Run the CONFIG utility to set up for a dual drive and you are all set. There is no need to reduce the size of CP/M to 44K from the 48K maximum as described in the *Commodore 64 CP/M Operating System User's Guide.*

Unfortunately, these advantages do not extend to software packages which only recognize the existence of the 1541 single drive. These programs presume that two drives will exist as device numbers 8 and

Continued on page 61

MATH DEFENDER

for the C-64 and VIC 20

By Andy Van Duyne

Math problems are falling from the sky! You, as the lone defender, must position your cannon and blast the problems with the right answer. If just one of the problems lands on your base, all is lost!

This math game allows the user to pick from addition, subtraction, multiplication, or division, alone or in any combination. Three skill levels (related to the speed of the problems' descent) are available. Two versions are included here, one each for the VIC and the 64. VIC players must ward off three problems at a time, while 64 users battle five! Points are awarded for each correct answer, and are deducted for each 'miss'.

Other than the number of problems falling, the only major difference between the two versions stem from the different sound addresses and screen widths. The title page is also a little fancier for the 64, and is located in a subroutine at the end of the program.

When the game is loaded and run, the title page will be followed by a menu of problem types. Pressing "Y" will include the indicated type, and 'N" will exclude that type. The next menu determines the skill level—the more advanced your skill level, the faster the problems will fall. The "F" key is used to move the marker ("cannon") to the left, and the "J" key moves it right. I experimented with several combinations of keys, and found that these particular two provided the best compromise between speed of moving the marker and protection from accidental number entries. The hands are centered on the keyboard very nicely. Other keys can be used, if you wish, by changing the values in line 109. Once the marker is situated under the desired problem, the user can enter the answer, followed by the RETURN key. There is no delete function (sacrificed for speed), so typos must be entered as incorrect answers. An incorrect answer results in a short buzz, and play continues after the score is decremented. Correct responses cause the problem above the marker to be "blown up", and a new one will take its place at the top of the screen. (Note: once an answer is begun under a particular problem, a RETURN must be pressed before the marker may move again—you cannot leave half-finished answers lying about the screen!) Play will continue until one of the problems make it to the baseline unsolved.

There are a few other aspects you may wish to modify to suit your needs. The speed of descent for each skill level is determined by lines 1500-1504, with the equation in 1504 actually setting the value. You can also adjust the parameters of the math problems in the routine in lines 40-48. The problems are derived from two seed numbers generated in line 42, which creates two numbers in the range 0-9. Actual problem formation for the types is handled by the following lines:

type	line(s)
addition	43
subtraction	44
multiplication	46
division	47, 48

Entry of either of the programs is rather straightforward, with no special typing problems involved. However, pay particular attention when typing the lines to establish the placement strings (14-17). If you prefer not to type either of the programs, I will make a copy for you. Send $3.00, a blank cassette or formatted disk, and a self-addressed stamped mailer to:

Andy Van Duyne
Box 226
Norwood, NY

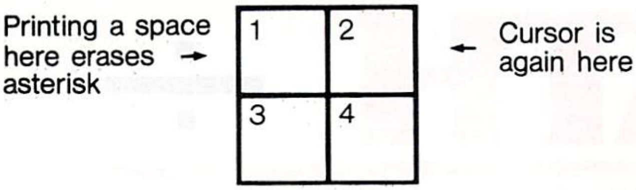
Please be sure to specify which version you would like (VIC or 64).

SEE PROGRAM LISTINGS ON PAGE 72

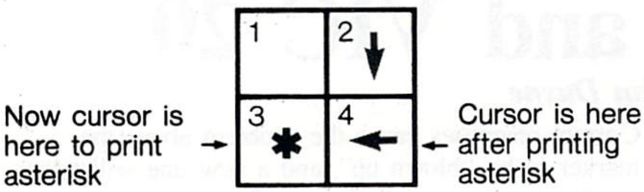
TRAINING YOUR CURSOR

Continued from page 38

Now we print the space, which will erase the asterisk and position the cursor in box 2 again.



We want the next asterisk to appear in box three. In order to get the cursor there, we will print a cursor down (box 4), a cursor left (box 3), and then print an asterisk.



The cursor is now where we want it to repeat this sequence again. In the program below, a FOR... NEXT loop will repeat this procedure 20 times to move the asterisk down the screen.

```
10 PRINT"{SC}{CR}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}";
20 FOR X = 1 TO 20
30 PRINT"*{CL}";
40 FOR Y = 1 TO 40 : NEXT Y : REM
 DELAY LOOP
50 PRINT" {CD}{CL}";
60 NEXT X
70 END
```

Using the techniques we have described here, you can move characters in any direction on the screen. With a little creativity, you will achieve impressive results. □

BRISK

Continued from page 41

in the Trump spot on the screen. Trump is really the suit of that card. If the suit of that card is hearts, all hearts are Trump for that hand. Trump can beat any card except a higher Trump card.

Each round consists of one play by each player. The player who leads sets the suit to be followed. The player who makes the second play must do one of the following things to win the round:

1) Play a higher value card in the same suit
2) Play a trump card

If the second player can't do any of the above he should leash. To leash is to play a card that has no point value. It is a good strategy to try to let the computer lead, so you can pick up points when it leashes. You might have a big point value card in that suit in your hand that you can play to win the round.

Brisk is a totally random game; no two games will ever be the same! *Brisk* has a 75% win ratio. Do you think you can do better?

For readers who are also programmers, here are some of the program's more interesting aspects:

Line 6	Sets the screen colors
Line 7	Sets the arrays up to hold the value of each card
Line 11	Turns on S.I.D. (Sound Interface Device)
Line 50	Sets all the values of DIMC$—to null values. This array is used to hold the cards that are played in each hand
Lines 100-143	Deal the first seven cards of each hand
Lines 150-250	Deal the cards during the hand; also check to see what cards have been played
Line 200	Randomize a value which represents a card
Line 210	Checks to see if this card has been played; if so we go back and get another number
Lines 230-240	Determine who won the last round and who should be dealt first
Line 247	Check to see if the last card in the playing deck has been dealt
Lines 300-330	Determine end of the first hand and start of the second
Lines 1000-1030	Read in the values of each playing card
Lines 8000-8999	Enable the player to play a card
Lines 10000-10999	The complete brain of the computer. From this subroutine all the plays the computer can make are determined
Lines 12000-12650	Determine who won the last round played
Lines 20000-21110	Give the points to whoever won the last round
Lines 25000-30600	Set the values of who won the round so that on the next deal the player who won will be dealt first

Those readers who don't care to type in the program may send $5.00 and a stamped, self-addressed envelope to Joseph Fusco, c/o Icicle, 1474 Hylan Blvd., Staten Island, NY 10305. A copy of the program will be sent promptly. Specify whether you require disk or tape! □

SEE PROGRAM LISTING ON PAGE 76

IFR (FLIGHT SIMULATOR)

Academy Software
C-64 (disk), VIC 20 (cartridge); joystick

"Pilot to bombardier! Pilot to bombardier!" Er, ah—no—this *is* a Cessna isn't it?

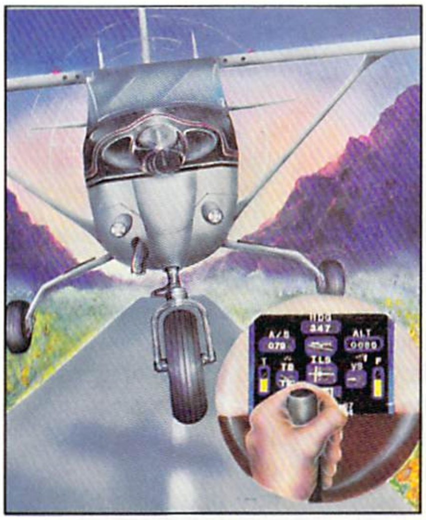
IFR Flight Simulator is not a game. It is a simulation of flying a small plane under Instrument Flight Rules (IFR). That means the graphics consist of a full-screen instrument panel and the controls are much more involved than pushing a joystick. Actually, you could almost call this program a flying tutorial. All that is missing is a detailed explanation of flying (which you could probably find at the local library).

So, if you are a budding young pilot, or if the guidance counselor said you're really cut out to be a smuggler, let's fly.

Once the program loads, you see all the instruments used for IFR flight—artificial horizon, altimeter, compass, tachometer, fuel gauges, glide slope localizer, and indicators for airspeed, vertical speed, turn and bank, flaps, and landing gear. You will also appreciate the LORAN navigation system which will show your coordinates. If you do not know what to make of all this, read the fourteen pages of single-spaced instructions. Before you manage a landing (as opposed to a crash and burning), it will all make sense.

The object of *IFR Flight Simulator* is to fly to each of the four airports shown on the main map. Five detailed maps will help you figure out approaches and routes. With nine different skill levels and elapsed time to worry about, you will put in many hours mastering this program.

At higher skill levels, there will be much more wind to blow you



offcourse. You will also have to land at a lower rate of descent and airspeed. More difficulty? Set the wind turbulence. In bumpy air, the instrument panel on your screen will really start to hop around.

When you get proficient at standard takeoffs and landings, you can try the mountain pass. I made it through once, but I used so much fuel setting up that I crashed on the other side. Although you are not allowed to do aerobatics (the artificial horizon goes wacko), you can go into spins. Is that vertigo I feel?

Since this is a real time (almost) simulation, you will have plenty of time to play with the controls which include a joystick and fifteen different keys. Operating throttle, nose trim, landing gear,



IFR's *graphics include a realistic full-screen instrument panel.*
READER SERVICE NO. 1

flaps, brakes and steering (on the ground), and tuning the instrument landing system and automatic direction finder only gets hectic in the last moments of a landing approach.

If you do not have time for the full flight between airports, you can use the program's editing capabilities to put yourself anywhere on the map, flying in any direction you want. Just be careful to give yourself an airspeed above the stall speed and an altitude above the local terrain. Gets messy otherwise.

The plane has a slow, kind of mushy response to the controls. This is definitely not like chasing Klingons. If you have banked left from your course, you will have to bank right exactly the same amount to get back on a line parallel with the original.

IFR Flight Simulator is a wonderful learning device for would-be pilots. Even my pilot friends, for whom this was nothing new, enjoyed it. Not a game, however, this program will send you back to study the instructions several times. The documentation states that this is not a substitute for real flight training, but it can certainly help you become comfortable with a lot of the basics.

Academy Software, P.O. Box 9403, San Rafael, CA 94912.

—Richard Herring

EARLY GAMES FOR YOUNG CHILDREN

Counterpoint Software, Inc.
C-64, VIC 20
Disk, cassette; keyboard

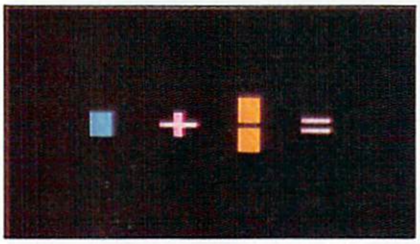
Moms, take note! Absolutely and positively the only thing wrong with this preschool program is its boring name! Even I, who usually cast a jaundiced eye at such glamorous-sounding games as *Spider Attack* and *Galaxian Explosion*, found myself sitting

down to review *Early Games* with less than enthusiasm. It sounded so blah! But within minutes I was delightfully surprised at this clever and practical program that will be a real treat for the computer fledgling in your home.

There are nine different educational games, introduced by (at long last for the four year old who can't read!) a Picture Menu. The menu consists of a series of pictures representing each of the different games. All the child has to do is press any key and *voila!* he's ready to play. There is also an articulate pamphlet that accompanies the program.

Here's whats in store for your youngster. He'll match big bright numbers displayed on the monitor with the proper key on the computer. In another game he'll count the colorful blocks displayed on the screen and press the correct number. There is a game in which he will add, and one where he'll subtract numbers between 1 and 9. He will also learn to identify shapes, determining which one of four shapes displayed is different. Cute graphics, too!

Having mastered all that, it's time for alphabet soup, where your child has several different ways to help him learn his ABC's. He'll match large colorful letters on the monitor with the identical letter on the keyboard. There's a game designed to help children learn the order of the letters in the alphabet. For example, the letter "A" is shown, the computer asks for the next letter, and so it continues through the entire alphabet. If he answers incorrectly, the computer displays the alphabet up to the letter in question. If the next response is also incorrect, the computer then flashes the correct answer. The added bonus is that as he learns



Addition and subtraction exercises.

the alphabet he also learns the keyboard. True, he might not be typing 60 words a minute at the end of the game, but with practice, who knows?!?

There's surely one thing he'll be able to do and that's to type his own name. All mom or dad has to do is enter the child's name in the computer. The monitor will display the name in nice large letters, and the computer then asks the child to type in his name. This may be a piece of cake for Ann or Tom, but think of Geoffrey or Anastasia! Kids love this game, and why not? Do you know any red-blooded American child who doesn't like to see his name in print?

The grand finale of *Early Games* is my out-and-out favorite. It's called Picture Draw, another less than scintillating title. But that's exactly what the child does: draw pictures by pressing different keys. The position of the key on the keyboard determines which direction is drawn, i.e., upper keys draw up, corner keys go diagonally, etc. By pressing the space bar the child can even change colors! It's a barrel of



Which shape displayed is different?
READER SERVICE NO. 2

fun, and I enjoyed it as much as my children did. Granted, it might not be as educationally enriching as the number and letter programs, but it's bound to excite your child to the limitless plusses and uses of a computer.

—Valerie B. Tamis

THE GAME SHOW

Advanced Ideas
Commodore 64
Disk

The Game Show, a clone of the old TV series Password, brings the thrill of the popular show back into your living room. Even better, it offers an entertaining way to introduce your children to a wealth of factual information.

Just to refresh your memory—those synapses get weak after so many hours in front of the tube—Password is a game for two teams of two players each. The moderator shows one player on each team a target word. That player then gives short clues to his partner. Teams alternate turns until one of the partners guesses the target word.

That is a description of *The Game Show* too, only the parts of the moderator and the clue-giving partners are played by animated characters on your computer screen. The two computerized players—Joe and May, the instructions call them—have real personality. At the beginning of the game the moderator introduces you to your onscreen partner who waves at you. From his raised podium, center stage, the moderator then turns and prompts each player as Joe and May take turns giving clues. Clues appear in cartoon bubbles above the players' heads while their lips move. If you guess a correct answer, your partner waves his arms with joy.

When you run the program, it

will ask for your name and the name of a second player if there is one. Then it asks if you want onscreen instructions. Read those and pick one of the subject areas for your game. Then the moderator comes on and, in those little bubbles above his head, gives you instructions on playing. Advanced Ideas has this instruction business down pat.

So how is this "game" educational? Basically, it is heavily into paired associate learning. In other words, if you want a kid to learn facts about a historical event or scientific phenomenon, you build these facts into the game as clues. The subject areas that come with *The Game Show* are advanced vocabulary, animals, algebra, computer terms, sailing terms, constitution, nursery rhymes, biology, famous cities, and airplanes. There are also six categories of words that start with the same letter. Advanced Ideas promises to make additional subject matter disks available (for $20 a crack).

If you do not like these categories, build you own. *The Game Show* comes with an authoring system. Not to worry—this requires no programming. It is as easy to use as the instructions. You can format a blank disk, create new subjects, and change or delete old subjects. When you create a new subject you must enter the target words and as many clues as you choose for each one. Enter the clues in the order you want them given; the program will not scramble them.

One aspect I really like is the program's ability to accept alternate target words. If you want both "krypton" and "Kr" to be correct answers in your chemistry game, type them both in. When you change an old subject area, you can delete any one or more target words, edit targets and clues, add new targets, add new clues to old targets, or list the whole shebang to the screen or printer.

Easy as this authoring system is to use, I have to admit that it is a bear to write new subjects. I wish that Advanced Ideas would author a long list of subjects and sell one-page printed copies for a buck apiece. Then we users could order the ones we want and type them in.



Employs paired associate learning.
READER SERVICE NO. 29

The Game Show is one of those pieces of educational software that are deceiving because they're so much fun. By manipulating the difficulty of the subjects and clues, you can make this program appropriate for any age group. The presentation, although it is a single graphics screen is appealing. And you can really learn by pairing descriptive facts with an object or term. At $40, however, this is not as good a buy as Advanced Ideas' other educational program, *Wizard of Words,* which includes five games on one disk.

Advanced Ideas, 2250 Ninth Street, Suite 104, Berkeley, CA 94710.

—Richard Herring

THE VOYAGE OF THE MAYFLOWER
Cadmean Corporation
C-64
Disk; keyboard

The Voyage of the Mayflower is a simulation, complete with 17th-century music, of the pilgrims' trans-Atlantic journey. Other than an 11 x 16 map and some diagrams on the back of the box, all the documentation is in the program. When you start the game, you'll read two screens of historical background. Then you will get to pick one of five skill levels or review a list of 100 terms relevant to the simulation and a bibliography of five books and two articles on the history of the voyage. The word list is just that—a list. The words are not defined, or even grouped into classes (like, the types of sails).

Each level of *The Voyage of the Mayflower* introduces new factors which the player must consider. Level 1 could have been left out of the game for all it's worth. Your ship appears on a map of the Atlantic with the two Plymouths clearly indicated. Type "go" to set sail. Your ship jumps a short distance westward. Type "go" again. Another jump. Eventually you hit North America (or sail under North America) due west of where you started. No heading commands are accepted. At any point, type "land." The game ends, with no score, and returns you to the main menu. Here is a whole level of the game just to teach you how to type "go."

Levels 2 through 5 are more entertaining. After the instructions at each level, you load the Mayflower to its maximum tonage. In level 2 you just load food, but by level 5, you are juggling weights to get the best combination of fresh food, salted food, preserves, water, spirits, five kinds of animals, three different feeds, stores and cargo. During the voyage, you can ration food and water or eat

the animals. After ten days or so without rations, the crew will be too weak to sail further.

Once the Mayflower is loaded, you will get a status display. It shows your remaining supplies of food and drink, the ship's location in longitude and latitude, the direction and knots of the current and wind, your ship's heading and hull speed, and your days at sea and miles sailed. The twelve sailing commands you can type in are also given. During most of the game, you will see a map of the blue Atlantic bounded by the green borders of North America and Europe/Africa.

Experiment with different headings to gain the maximum benefit of wind and current. And avoid the North Atlantic storms; the screen will flash as the sails of your ship are blown out. Success can only be guaranteed if you learn where the Sargasso Sea, Gulf Stream, and other major currents are.

You are encouraged in *The Voyage of the Mayflower* to play with variables as much as you want. Only when you type "go" does the time pass and the food diminish. When you finally type "land" you get a score based on how much food and drink you had left, how many days the crew was on rations and how many degrees you were from Plymouth. If you type "land" even when you are in the middle of the Atlantic, the game will end, giving you a huge negative score for being so far from Plymouth.

The goals of this educational game are good, but it is hard to say that they have been met. If you get caught up in the game, you will certainly be encouraged to learn about history and sailing, especially in the Atlantic. But a child beginning with Level 1 may become bored. There is no easy way to get complete instructions and the graphics quickly become uninspiring. The word list is nice, but what are we to do with it? In a classroom, maybe the teacher could structure some activities, but that is a lot to ask of a parent. While a child certainly could learn from manipulating the variables in this game, I doubt that many kids will find it interesting enough to pursue on their own.

Cadmean Corporation, 309 Koch, Ann Arbor, MI 48103.

—Richard Herring

C.A.R.S.

New Leaf, Inc.
C-64, VIC 20 (+24K)
Disk, Cassette

Whether you own one car or a fleet, *Car Analysis Reporting System (C.A.R.S.)* will enable you to analyze your spending on each car for business or tax purposes, tell when tuneups and overhauls are due, or just keep the best records on your block.

C.A.R.S. is actually a set of programs. Backup lets you easily and quickly (in about eight minutes) make a copy of your *C.A.R.S.* disk. Don't break out your eyepatch and wooden leg just because this program is easy to copy. It also comes with an ingenious little device that plugs into either joystick port; without it the program won't run.

All you aspiring software authors, pay attention. This program's manual is excellent. A useful table of contents is included, as well as a complete index. The first chapter is a tutorial; next comes a step-by-step description of each feature, in the same order as the program's menus.

You enter expenditures in six categories—fuel, maintenance, tuneups, depreciation (or car payments), taxes, and insurance. Refuelings must be entered in order. *C.A.R.S.* will not let you create a new record with a lower odometer reading or an earlier date than the previous record. If you enter only the month and year of a refueling, the program will prevent you from entering any other data for that month.

You can go back later and look at each entry. Find an error and you will have to delete the entry and type it in again. Be especially careful with depreciation, taxes, and insurance. *C.A.R.S.* rolls all your entries into one category called Ownership. Enter one of these records incorrectly and you'll have to reenter them all—that is, if you catch your error. Since you cannot review the individual entries you made as depreciation, taxes, and insurance, you have no way to double-check your

data.

When your data is all in, you can produce nineteen different bar graphs on the screen or printer. Fifteen of these are different ways of looking at fuel consumption and cost. Why would you want all these charts? Got me. The manual gives uses for only a few. Averages are the best indicators of the need for a major overhaul, and rolling averages can show when your car needs a tuneup. Total cost per mile at each refueling is the chart you fleet owners will use to compare vehicles and determine when to replace them.

C.A.R.S. has several shortcomings I have not mentioned yet. No record for one car can have over thirty-six entries. When you enter more, the oldest entries are summarized by the program. If you buy gas once a week, your record won't last out the year. A major flaw is that once you have loaded a record, you cannot, even by following the directions, clear the data from memory.

C.A.R.S. comes with a separate depreciation program. You can run the value of any item through standard three, five, or seven year depreciation schedules and get values for each year.

There are several features about *C.A.R.S.* that I like. When you update a file, the program, rather than overwriting the old file, renames it and saves it as a backup. The menu program lets you see every file on the disk in alphabetical order and run one by pressing a function key. During data entry, the program will automatically convert between liters and gallons. And most commands require only a single keystroke—you won't wear out the <RETURN> key.

Still, I cannot justify this program. If I decided to keep detailed records, I would use a good database program. Then each item changed in a tuneup or repair could be listed individually. True, that would make it harder to get all those nifty graphs.

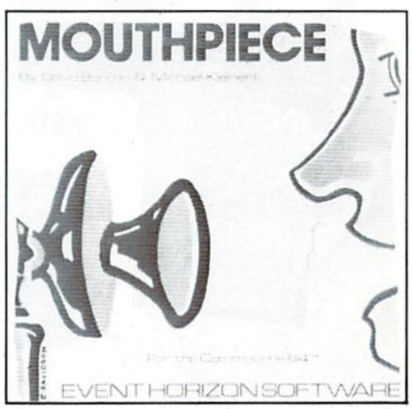
C.A.R.S. really seems like it was developed for business use. If you have just one vehicle, will you take the time to run the program every time you buy gas or a new set of wiper blades? Maybe you should use that time to tune up your car. If yours is already tuned, you can tune mine—it's perpetually in need of work!

—Richard Herring

MOUTHPIECE

Event Horizon Software
C-64
Disk

If you own a VIC or a 64 and a modem, you have probably enjoyed some successful telecomputing. Most likely you used the standard Commodore software, *Term 64,* or another "dumb" terminal program. This type of program allows you to send and receive information, but does not permit you to tap the other capabilities of your computer.



READER SERVICE NO. 31

"Smart" terminal programs, those which take advantage of your computer's capabilities, add a new dimension to telecomputing. The best of the "smart" programs has arrived in a nifty package called *Mouthpiece* ($39.95).

Now your "modem moments" can be magically enhanced by a terminal program with many powerful and easily obtained features.

After loading *Mouthpiece* you are greeted by a healthy menu clearly displaying thirteen lettered options, each quickly engaged by striking the Commodore logo key and the appropriate letter. Features include: color selection, duplex toggle, macro definition, cursor select, data parameters, and audio parameters. A time-saving advantage built into the program permits menu commands to be called directly from terminal mode.

The terminal screen presents an inverse bar running across the top of the screen. Called the "status bar," it is likened to a dashboard for the *Mouthpiece* driver. The bar displays the duplex setting, full or half, status of the review buffer, and a running indicator showing the exact number of bytes of memory available for capturing or downloading text or programs, about 30K.

Additional functions are provided for control of the *Mouthpiece* review or capture buffer. This feature is what permits you to download—bring text and programs into your computer's memory and then either save them to disk or print them. Once a program has been received, a simple command allows you to "KRUNCH" it into a program file. Normally, the job of then creating text files for programs you wish to upload becomes a chore. However, *Mouthpiece* lets you bring the program file into your 64's memory with a "Load Buffer" function. Then, by hitting a key designated "Create Text File," the text file is automatically produced and ready to send via the "Send Buffer" command.

With a 64, a modem, and *Mouthpiece,* your computer becomes a smart terminal and you become a potent and capable user. The storehouse of knowledge, trivia, and news available through information utilities is yours to receive, save, send, and enjoy. Why make a smart machine dumb? *Mouthpiece* unlocks the potential of your computer and modem.

Event Horizon Software, P.O. Box 1327, New York, NY 10028. Phone number is 212-535-0697; Compuserve user ID is 72436, 1456.

—Kenneth C. Holvig

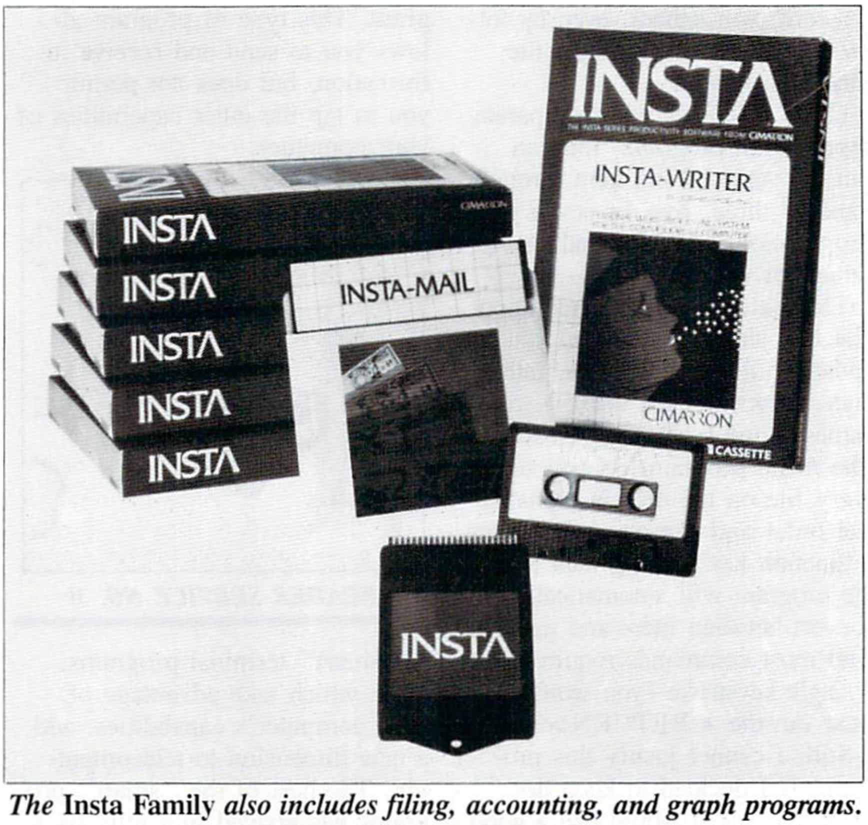
THE INSTA-FAMILY

Insta Software
C-64
Cartridge, disk

Five programs make up the *Insta* family of applications software, which rarely lives up to its name. *Insta-Writer* is the "papa," a cartridge-based word processor without which you can't use *Insta-Mail* or *Insta-File.* The best thing going for *Writer,* unfortunately, is that you don't have to wait for it to load from disk. It's a survivor from the Stone Age of C-64 software, a line-oriented program that displays a number on the left side of the screen for each line of text. A maximum of 62 lines are permitted in RAM, the equivalent of a single typewritten page.

After you've filled a page, it must be saved before proceeding to create page two. The second page is then saved as page two of the same document. When it's time to print, you have to load page one and print it, then load page two, and so on. This makes *Insta-Writer* extremely inefficient for people writing anything longer than a letter to the manufacturer to demand a refund.

In all fairness, it's easy to use. Control-key combinations and the



The* Insta Family *also includes filing, accounting, and graph programs.
READER SERVICE NO. 30

function keys facilitate common text-editing operations in a logical manner. Cut and paste operations are executed by indicating the numbers of the lines to be manipulated, and a text buffer allows you to move or copy text from one page to another, even though you can only have one at a time in memory. But you can't embed control codes to coax underlining, boldface, and other capabilities of your printer; there's *no* Search and Replace; and line spacing is not adjustable. Give me a number two pencil and a napkin from Mulligan's Bar any day of the week!

Insta-Mail is on disk, and loads automatically when you press "control-5" from *Insta-Writer's* main menu. Creating a mailing list is foolproof: just fill in the six blank lines that are prefabricated to hold last name, first name, address, city, state, and zip code. The only problem lies in the fact that each line can't hold more than 21 characters. If a street name happens to be longer, or an address calls for more than six lines, forget it. Most mailing list programs allow you to determine the number of lines and their length. This one deprives you of that freedom, trading it off for a program whose files don't demand any advance planning and are easy to create and edit.

Any of the six lines of information may be inserted into an *Insta-Writer* letter by typing in an up-arrow followed by the number of the desired line. This facilitates churning out personalized form letters. (You can tap the space bar to move back and forth between the mailing list and a document currently in the word processor.) It's also possible to sort addresses alphabetically on any of the six lines, and to select specific addresses for insertion rather than the entire list. Mailing labels can be printed in a one-up format.

Insta-File is also on disk. It's simple to create and enter files and records with this program, but it also restricts the user in practical applications. You can't name a file—instead, each must be dedicated to a specific disk. This means that the file on your tax records and the one on your car records will be on different disks, even though they may both be small enough to fit on one. The result? You wind up using more disks than necessary.

Each file is preconfigured with records containing 63 fields. (You don't have to use all 63 in a file.) The snare is that each is set up to hold a maximum of 25 characters, so you can't fit as much information into any specific field. (Fields in most databases hold up to 250 characters.) Searching a file for specific records is inefficiently implemented and often slow. At least you can insert data from any fields into a document created on the word processor, accomplished in the same manner as with *Insta-Mail.* Reports may be generated, even drawing upon bold, underline, and other printer features. The main problem with *Insta-File* and *Insta-Mail* is their dependence on an inferior word processor, and this side of the "family" suffers from severe genetic defects.

Insta-Calc is a distant cousin that offers one distinct advantage —it interfaces with *Insta-Graph* to convert spreadsheet data into charts and graphs that can be displayed or printed. (*Calc* is a cartridge/disk package, and *Graph* comes on disk.) Even the infinitely superior *Multiplan* can't match this feat. Unfortunately, *Insta-Calc*'s spreadsheet will accommodate more than 30 vertical rows and 20 horizontal columns, making it impractical for large budgets. Only people desiring to plan their household budgets and small businesses with teeny, tiny businesses will get their money's worth from this program. The functions are limited to exponentiation, summing, addition, subtraction, multiplication, and division. Also, recalculation (after changing a number anywhere in the spreadsheet) is not automatic. You must manually request this operation, and perhaps jog around the block while waiting for it to be completed. Typical features such as copy, replicate, insert and delete rows and columns, and printing are standard, but unless your financial applications demand a miniscule workspace and you can't survive without *Insta-Graph*, keep your distance from this duo.

The documentation for the entire series is uniformly deficient. Each program's features are described in detail, but in an unorganized manner and without true tutorials to gradually step you through the learning process. If you're already familiar with how each type of program works, you should be able to decipher the documentation. But, with so much *quality* software available elsewhere, why bother?

Insta Software, 2158 Hathaway Street, Santa Ana, CA 92705.

—Shay Addams

SUPERBASE 64

Precision Software, Inc.
C-64
Disk

Designed for serious business users, *SuperBase 64* is based on a dual-control system: beginners can stick to the function keys listed on a pair of main menus until confident enough to enter commands directly into the command line at the top of the screen. (The latter

method saves time by allowing you to enter a series of commands at once.) Incorporating numerous advanced features seldom seen in a database manager for the C-64, the program provides capabilities that can gradually be tapped as you gain familiarity with the fundamentals.

Unlike many databases, this one doesn't ask you to specify the length of fields or number of records when setting up a new file—just type in the field names exactly as they should appear in the records. At the beginning of each field, press a key to indicate the type of field (text, key, numeric, result, constant, or date), then cursor horizontally until the desired length is reached and hit return. Each field can accommodate up to 225 characters, and a record may fill as many as four screens. For result fields, whose contents are automatically calculated on the basis of numeric data in other fields, a formula is typed into the command line. The appropriate field names are employed in this process, as "[Price] - [Deposit]" for the field titled Balance. The border, background, and text color of each file can be adjusted, with a different scheme for each file if you choose.

SuperBase furnishes a full complement of methods for conducting searches. Records can be located by specific key fields, or by looking for matching data in one or more other fields. "Greater than" and "less than" symbols come in handy when retrieving records based on data in numeric fields, and the equivalent of BASIC's AND and OR functions are available. Numeric and date fields may also be searched for matches within a specified range. All searches are initiated by pressing a function key to enter that mode. A sub-menu shows the options, which include moving directly to the first, last, previous, or next records, as well as adding, deleting, or updating a record. The "Match" option, used more than any other, involves typing the search criteria into the selected fields on a blank form. The first matching record appears, and you press "m" to see the next one. Whenever a record is on screen, a line of letters runs across the top. These are abbreviations for the commands on the search submenu, which can be executed by pressing the corresponding letter.

An unusual and expedient option called "Find" displays a list of the key fields of all records matching the search criteria. The list is automatically saved, and you can use the "Output" option to display or print all or selected fields from the records in the list. Commands indicating which fields to output and how they should be formatted are typed in on the command line.

The ability to convert database records into a sequential file that can be read by a word processor is not uncommon, and *SuperBase* includes an Export feature for this purpose—and an Import option that will read a sequential file created by another database program. This means you can transfer existing files from your old database without having to key them in one at a time.

One reason business users should consider upgrading to this system is that *SuperBase* is far more than just a filing system—it's actually a database programming language that's completely accessible by the user. Some BASIC keywords have been replaced by others devised especially for manipulating and managing database records and files. The "Program" option makes it possible to implement this language in a program that automatically executes an entire sequence of operations you regularly carry out on your files. You can even create your own menus and redesignate the function keys' operations. A library of programs can be saved on disk, then loaded individually into *SuperBase* when needed and executed by pressing a single key. One of the new keywords, "link," establishes a link between any two files in a database (each database holds up to 15 files), and an example listing shows how to construct a program that will coax records in one file to automatically update those in a *different* file. This is the closest thing to *dBase II* currently available for the Commodore 64. Naturally, a foundation in BASIC reduces the amount of time required to get the most out of the programming option.

Still not satisfied? There's a feature called "Calc," which performs mathematical operations or BASIC functions on information stored in numeric and result fields of the current record. This facilitates the equivalent of doing "what-if" calculations on your database. "Batch" does the same thing for an entire file, or only those records selected from a key list predefined with the "Find" option.

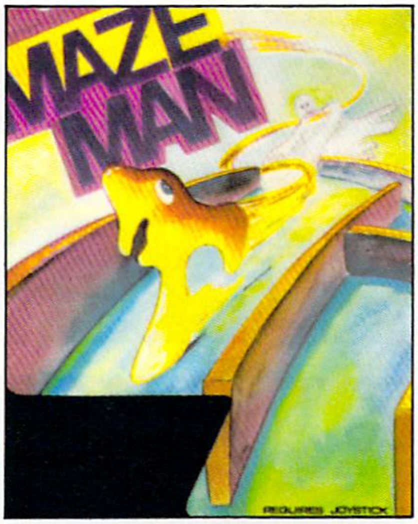
Error-trapping is excellent. Files are efficiently backed-up with a copy-disk option, and other standard file functions can be executed from the "Maintenance" submenu. The looseleaf, vinyl-bound manual includes a thorough tutorial, reference section, introduction to programming (with examples) and appendices. This wide range of features contributes to the long-term value of this program, and anyone planning on harnessing the C-64 in an office or business environment can't go wrong with *SuperBase*. ***—Shay Addams***

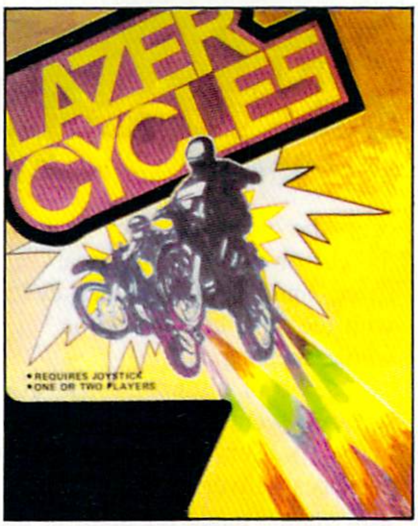
FALCONIAN INVADERS
LAZER CYCLES
MAZE MAN
ULTREX QUADRO MAZE
TYLER'S DUNGEONS
Creative Equipment
C-64
Cartridge; joystick

Judging from the packaging of these games, as well as the games themselves (which are sold separately), Creative Equipment must be a new and relatively low-budget company. In that situation, a company should make up in imagination and daring what they lack in experience or finance. Looking at Creative Equipment's games, all sorts of scenarios spring to mind: enthusiastic young software designers gone cynical and lazy because of overwork and underpayment; imaginative programmers crushed under the heels of unimaginative management; or, at the very least, a stubborn insistence on sticking to a game plan conceived at 1 a.m. on low blood sugar. I don't know what went on at Creative Equipment, but the results are derivative and second-rate.

Falconian Invaders is the most fun of the lot, given my preference for space games over maze games. It's a poor man's *Buck Rogers—Planet of Zoom*. The view is from behind a space ship. Using a joystick or trakball, the player blasts away at alien ships which appear on the horizon and lurch and sputter toward you. You win points and extra fuel for hits. A tracking option can be selected to help you aim. *Falconian Invaders* boasts graphics that a VCS designer would be ashamed of, including a feeble attempt at a *Buck Rogers*-like scrolling effect. If you like to aim and shoot, you might enjoy this one.

For players who enjoyed *Surround* (or the cycle sequence in the Disney movie, *Tron*), *Lazer Cycles* may have some appeal. The game is the same. One or two players each control a cycle which leaves a solid wall in its wake. On a clear field, the object is to force your opponent, human or computer, to crash into a cycle





READER SERVICE NO. 3

trail or the screen border. The player has the option of playing within a border or wraparound. In early rounds, play is rather slow; in later rounds action is so fast that it's almost impossible. Graphics are jerky throughout, but the music is excellent.

Maze Man is a polyester *Pac-Man*. The main character is Clyde, who resembles a *Pac-Man* ghost tied up for the laundryman. Clyde must avoid squids while gobbling dots strewn in a maze. When Clyde devours the occasional "power square," the squids freeze for a few moments; they can't harm Clyde, Clyde can't harm them. . . he can merely gobble dots unimpeded. Atari's *Pac-Man* is now available for the 64. 'Nuff said.

Ultrex Quadro Maze presents the player with no fewer than four mazes onscreen at one time (a hack publicist's dream, that). When the player has eaten the dots of one simple maze, he continues on to the next. When those four are completed, four more appear onscreen. And so on. Goblins make an occasional appearance to complicate gameplay, but not nearly enough. As you can imagine, the mazes are quite simple. Young children just might enjoy this one. In many ways (except play value) it's a hoot.

Tyler's Dungeons is as hard to dislike as it is hard to enjoy. Your character is sort of an upright ant with a golf club or a squid or something. You propel this whoozis through a series of simple mazes in search of treasure. While using an occasional elevator which takes you hither and yon, you avoid a Lurk (deadly to the touch), Cross Traps (rooms that will collapse on you), and Snakepits. The mazes here are carefully rendered; they have plenty of texture and color, but gameplay is oh-so-very simple. Dare I recommend this one to the very young children of America? No way. I could lose my kneecaps.

Creative Equipment, 6864 W. Flagler St., Miami, FL 33144.

—Dan Hallassey

KIDWRITER
Spinnaker Software
C-64
Disk, keyboard

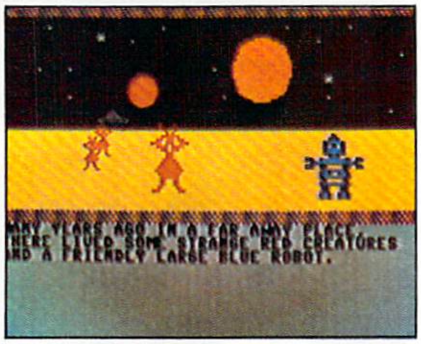
Kidwriter is an electronic storybook generator and word processor tutorial. That's computerese for: *Kidwriter* is an excellent program for children of ages 6 to 10 that allows them to create their own storybook pages and save them.

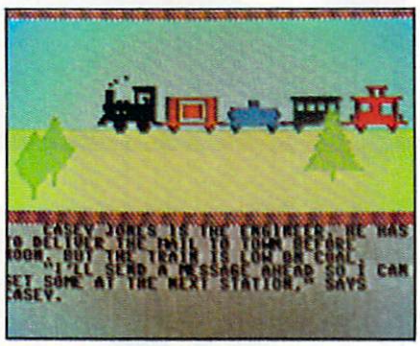
The program is all menu-driven. The first menu the child meets offers the options of making a new picture, retrieving an old one, seeing a catalog of old pictures, or quitting the program. If the child chooses to make a new picture, the second menu appears.

Menu two is the major work station of the program. First, the child is given the option of selecting one of ten possible backgrounds for the picture to be created. Each possible choice is displayed one at a time with a tap of the S key. Once the background is chosen, the child selects the objects and figures for the foreground. With each tap of the F key, one of the 99 possible figures is displayed. These include men and women—front and profile—aliens, vehicles cosmic and earthbound, animals, celestial objects real and fanciful, geometric forms, tools, buldings, and numbers and letters (for title pages).

When an object is chosen (with the P key), menu three is displayed. This allows the option of changing the color and size of the object as well as moving it around the screen with the cursor control keys. When the child is satisfied, the D key propels him or her back to menu two for more selection of figures or objects. When all the objects are chosen (there is a limit of eight or so per frame) and positioned, it's time to write the story.

Approximately two thirds of the screen comprises the picture. The lower third allows room for the child's text. *Kidwriter's* word processor has the fundamentals only. With the cursor controls and delete key, a portion of the text can be corrected, deleted, or typed





READER SERVICE NO. 10

over; the clear/home key brings the cursor to the first textual space, and the clear/home key with the shift key erases the whole shebang.

A sense of disappointment in software seems to be common nowadays. Very rarely do software packages deliver on the extravagant claims made for them in the past year. Objections are being made that educational/leisure software provides little more than has traditionally been offered in books, playing cards, boardgames and so forth. Seen in that light, *Kidwriter* is merely an electronic box of crayons that requires none of the motor skills encouraged by hand drawing and only a fraction of the craftsmanship and subsequent pride of accomplishment. Further, the program contains but 99 figures and ten backgrounds; that barely scratches the surface of most children's visual imaginations.

All true, and at the same time irrelevant.

Some children take to a computer like a duck to water; they become obsessed with it. Others may need some cajoling and still will never enjoy it. Many probably fall somewhere in between. Alert parents will recognize the developmental strengths and weaknesses of their child's natural inclination. They may find that the computer-obsessed child requires, at least initially, an electronic means to create visually or write precisely. They may find *Kidwriter* to be the ideal way of luring a suspicious child to the computer or, for those kids with a creative bent, an all-new medium for their genius. *Kidwriter* could be used to create elaborate flashcards or—youth of America, forgive me—spot quizzes.

Patience will be required to run through the figure menu; there is no way to select directly from the 99 choices. And simple as the word processor is, the parent will want to help the young child through its paces at first. In fact, the parents may want to hover a while longer. *Kidwriter* is an excellent program to stimulate interaction between parent, child and computer.

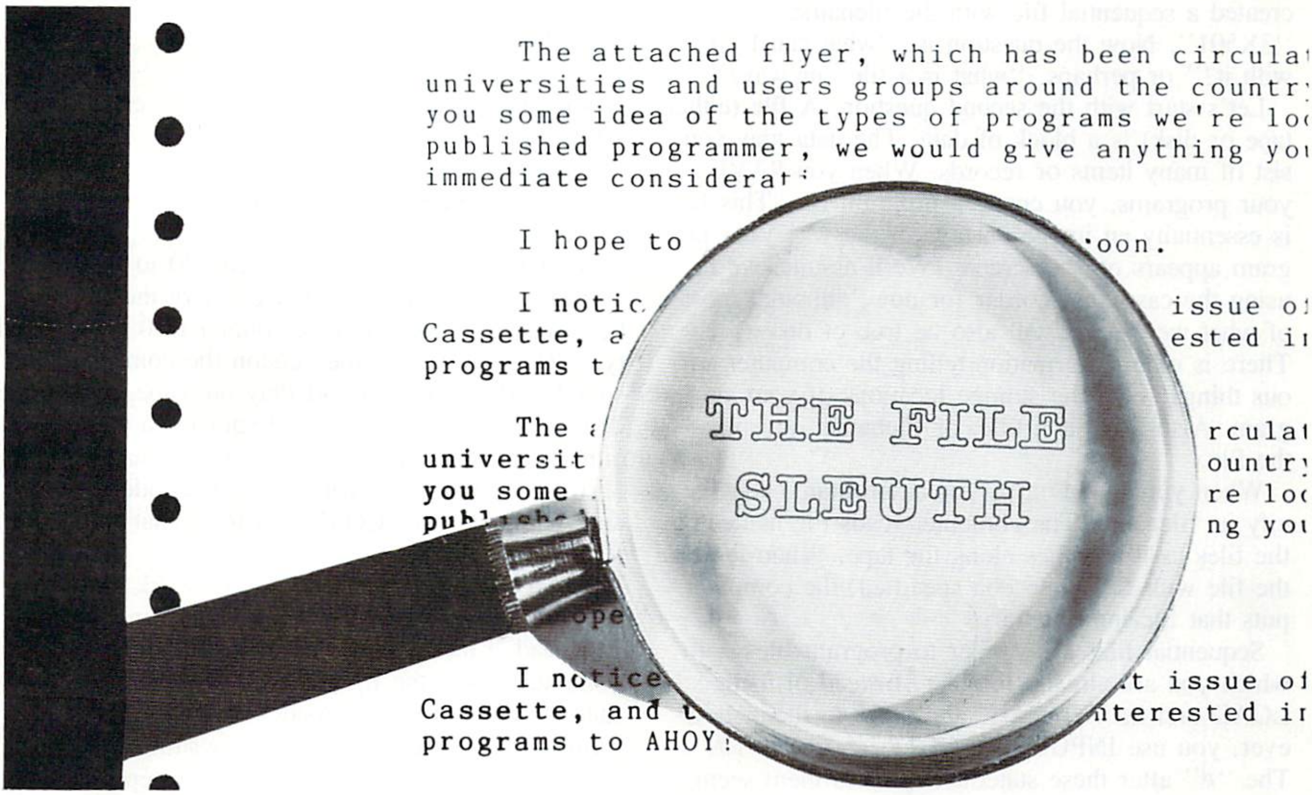
Spinnaker Software, 215 First Street, Cambridge, MA 02142.

—Tim Moriarty

UPCOMING AHOY! REVIEWS

Word Processor, Professional Version • C232 RS-232 Interface • ESP Calc • Script 64 • Clone Machine • The Heist

RUPERT REPORT



An Investigation of Sequential Files

by Dale Rupert

Your hand is shaking as you reach for the file folder on the desk. The room is dark, but you can clearly read the bold letters on the front: CONFIDENTIAL.

You open the file. Inside you read:

> You are to create a computer file (either tape or disk) by entering and running the following program on your Commodore computer. This file will have the name 3X501. It will contain information which has been encrypted according to directive E78-402SQ. The encryption code number for this file is 31, and of course that number is required for decoding the file. This information as well as the encryption code number must not be revealed. This folder must be destroyed after the folowing program has been entered into your computer.

```
1 REM >> CREATE CLASSIFIED FILE 3X501
5 INPUT "TAPE (T) OR DISK (D)";A$
10 IF A$="T" THEN 30
20 N=8:OPEN N,N,N,"@0:3X501,S,W":GOTO40
30 N=1:OPEN N,N,N,"3X501"
40 READ D$
50 PRINT#N,D$
60 IF D$="***"THEN CLOSE N:END
70 GOTO 40
80 DATA 118,100,107,98,110,108
90 DATA 100,63,115,110,63,115
100 DATA 103,100,63,118,110,113
110 DATA 107,99,63,110,101,63
120 DATA 98,110,108,111,116,115
130 DATA 100,113,63,101,104,107
140 DATA 100,114,64,***
```

Once you have run the above program, you have created a sequential file with the filename "3X501". Now the question is, "what can I do with it?" or perhaps, "what is a file, anyway?"

Let's start with the second question. A file (either tape or disk) is a block of data. The data may consist of many items or records. When you SAVE your programs, you create a program file. This file is essentially an image on tape of the way your program appears on the screen. (We'll assume we're using the cassette recorder for now, although most of what we discuss will also be true of disks.) There is other information telling the computer various things about the storage locations of your program. Also there is a title, or filename, attached to the file.

When you LOAD a particular program, you specify its filename. The computer reads the names of the files as it searches along the tape. When it finds the file with the name you specified, the computer puts that file into memory.

Sequential files are similar to program files with which you are already familiar. Instead of using LOAD and SAVE to read and write to them, however, you use INPUT# (or GET#) and PRINT#. The "#" after these statements makes them seem somewhat mysterious. We will see that looks are deceiving. These statements are really no more difficult to comprehend and use (after a little practice) than more familiar ones such as INPUT, GET, and PRINT.

The files we will deal with are called sequential files because of the way that data is stored in them and the way in which the data must be read from them. All tape files are sequential. Each data block or record is written right after the preceding one. The recording head in a cassette recorder certainly can't jump around to non-sequential places on the tape.

Similarly, all data must be read in a "single-file" fashion. The computer can't read the fifth record until it has passed the fourth record. Consequently, to access the fiftieth data item in the file, the computer must be instructed to read and count the forty-nine records in front of it. The sequential files are to be contrasted with random-access files, which exist only on disks. We will work only with serial-access or sequential files.

GETTING STARTED

Let's see what it takes to create a very simple file. This program stores the word "COMPUTER" in a file called "FIRSTFILE":

```
10 OPEN 1,1,1,"FIRSTFILE"
20 PRINT#1, "COMPUTER"
30 CLOSE 1
```

If you are using a disk instead of tape, change line 10 to this:

```
10 OPEN 8,8,8,"FIRSTFILE,S,W"
```

Also change the 1's in line 20 and 30 to 8's.

Position the tape to the place where the file is to be stored and reset the tape counter to 0. When you type RUN, with the tape version the computer responds "Press Record and Play on Tape." With the disk version, the drive should spin for a moment. In either case, the computer writes the filename "FIRSTFILE" along with some other identifiers, and then it writes "COMPUTER." That's all there is to it!

This certainly looks like a lot of work to write a single word to a file. The bulk of the program is overhead. Lines 10 and 30 are necessary to tell the computer about the file itself. Line 20 does the actual writing to the file. Aside from the #1 (or #8), writing to the file looks just like writing to the screen: PRINT "COMPUTER". Conceptually they are identical.

The OPEN statement tells the computer about the file we want to create. Just as there may be several open files on a desktop, we may open more than one file at a time in the computer. We identify each file by number, the file-number. The file-number is the first value in the OPEN statement. We opened file number 1 for tape and number 8 for disk. We could use any number between 1 and 255 although numbers below 127 are recommended. Once we assign a file-number to a file, any access to that file is made by specifying its number.

Since the computer deals with disks differently from tapes, we must identify the type of device on which our file will reside. The second number after the OPEN statement is the device-number. The cassette must have a device number of 1, and the disk must have a device number of 8. (Other devices such as the keyboard, video display, and printer also have specified device numbers.)

The third number after the OPEN is the secondary address. This number tells the computer whether we are reading or writing data to or from the tape (when the device number is 1. A "1" here means that we are writing to tape). A "0" would mean that we were reading from tape. A "2" could be

Continued on page 95

MSD DUAL DISK

Continued from page 46

9. To obtain the benefits of two drive operation with these packages, a second single drive as device 9 would be required.

The Duplicate command is used for making fast copies of an entire disk. This is actually only part of the capabilities available to users of a dual disk drive. Consider the Copy command, which is supported by the Commodore DOS. With a single drive, the results are of limited use. The copy ends up on the same disk as the original file. About the only useful feature of this command is for concatenation of files. Used in conjunction with a dual drive, the copy command takes on entirely new meanings.

Copy allows the transfer of files from one disk to another. In this case the destination disk must already have been formatted. Files from one disk can be copied to another disk which may already be partially full. For example,

```
PRINT#15,"C1:NEW FILE=0:OLD FILE"
```

copies OLD FILE from drive 0 as NEW FILE on drive 1. All the files from one disk can be easily transferred to a second disk by

```
PRINT#15,"C1=0"
```

This will copy an entire disk on a file by file basis. The process is considerably slower than the Duplicate command, but is faster and more convenient than the repetitive LOADs and SAVEs required with single or dual disk drives. The actual time for an entire disk depends on the number of files and the size of the individual files. As an example, the Commodore 64 Disk Bonus Pack takes about nine and one half minutes to COPY.

The biggest advantage of the DOS file maintenance commands is that no software has to be loaded in the computer to perform these tasks. Virtually all our single drive copy programs were obsoleted by the dual drive. Disk file maintenance can be easily performed from within any applications program capable of issuing DOS commands. Some of the timing tests for this review were actually done from within *Easy Script* while this review was being written.

Actual speed of operation on the serial bus was not improved by the MSD SD-2. The time required for a straight LOAD or SAVE operation was actually about eight per cent longer than with the 1541. This was not entirely unexpected. Data transfer over the serial bus is controlled by the computer and not the disk drive. The apparent slowdown of the MSD SD-2 was a deliberate engineering decision by MSD to ensure timing compatibility with all peripherals available for the serial bus. This is a wise decision on their part in view of previous compatibility problems experienced by Commodore and others.

There will be some significant speed improvement for applications involving random files and other multiple file access. The disk internal operations are significantly faster than for the 1541, as evidenced by the 17-second format time. We could not obtain exact figures on this speed improvement, but we do estimate as much as 20% for some operations. More significant results were obtained with the MSD CIE interface card. This provides an IEEE-488 port on the Commodore 64 for use with the MSD drive as well as other CBM peripherals.

The SD-2 did seem to be more sensitive to disk errors than the 1541; however, this may have just been a sample-to-sample variation. Sectors which came up bad on the MSD were marginal but readable with our 1541. Interestingly enough, all of these problems occurred with disks prepared on outisde drives. We experienced absolutely no pro-

blems while exchanging disks with any of our in-house drives.

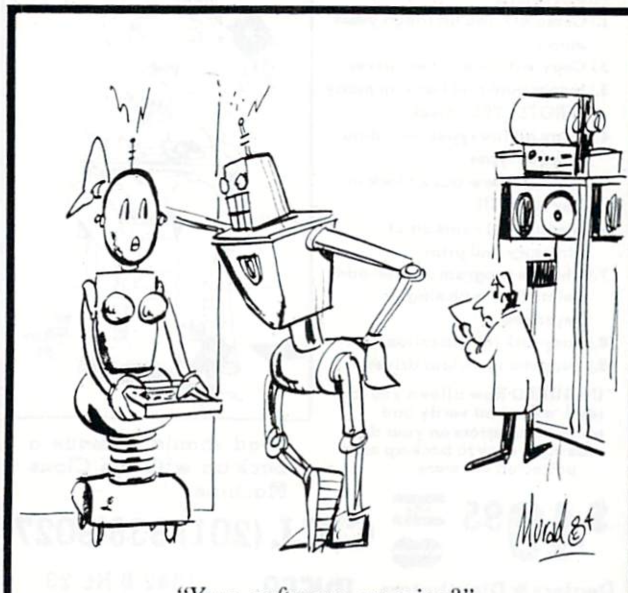
The drive's sensitivity was most noticeable with the Duplicate command. The proper operation of this function requires a perfect disk. Any errors or marginal sectors will abort the operation. A brief analysis reveals the reason. At 1 minute and 54 seconds for a complete backup, less 17 seconds to format the disk, there is only 97 seconds to perform the actual copy. At a drive speed of 300 revolutions per minute and 35 tracks to copy, there is not enough time for extensive error checking and reentries. In comparison, normal operations, for both the MSD SD-2 and the 1541 will make over thirty tries at a sector before reporting an error.

THE MANUAL

The manual supplied with the SD-2 is a model of clarity. All the standard DOS commands are clearly explained with examples for each. The explanation regarding the difference between the BLOCK-READ and BLOCK-WRITE commands and the USER1 and USER2 commands was particularly well done. However, the manual does manage to leave out some interesting points: for example, the DOS sequential file append capability. This command was left out of the 1541 manual as well. The syntax is:

```
OPEN3,8,3,"0:FILE ONE,A"
```

This would have the effect of appending all data sent to logical file number 3 to the end of sequential file FILE ONE. We tried this out with both the SD-2 and the 1541 and it does work.



"Your software or mine?"

Unfortunately the manual lacks disk layout diagrams. These drawings, included in the 1541 manual, contain useful information regarding the disk track and sector layout and the contents of the disk directory.

The SD-2 does not come with a demonstration disk. We felt this to be a significant shortcoming, particularly for users who elect the SD-2 as their first disk drive. Some kind of test/demo disk with sample programs and disk utilities, such as the DOS wedge, should have been provided.

RELIABILITY

In view of past experience with other hardware, this is a touchy topic indeed. We must report that the first sample of the SD-2 failed on the first day. Occurrences of this sort are associated with every manufactured product. Reliability engineers describe the early failures of a product as "infant mortality." This simply means that the most likely time for a factory defect to show up is very early in the product's useful life. All reputable manufacturers make allowance for these possibilities by offering some sort of product warranty.

In the case of the SD-2, MSD offers a six month warranty, twice the norm for this industry. In addition, MSD will turn around any returned units within 48 hours of receipt. We did receive a replacement unit posthaste and have been able to give the drive a thorough workout. A demonstration session at one user group meeting produced over 40 copies of *Comal* in an hour and a half. We estimate that the intensive use during the one-month test period prior to the writing of this report was approximately a year's throughput for a 1541 disk drive.

CONCLUSION

The Micro Systems Development SD-2 Super Disk Drive is a sturdy, well-designed product. It makes available to users of the Commodore 64 and the VIC 20 all the file handling capabilities long taken for granted by users of the PET and CBM line of computers. Serious users of the Commodore 64 will easily be able to justify the $695 price tag. The time savings alone, when backing up disk files, will rapidly pay for the machine. The drive is suitable for duplication of distribution software by program developers, user groups, etc. The inclusion of both the VIC serial bus and IEEE-488 bus make the SD-2 a welcome addition for Commodore users and a suitable replacement for the CBM 4040. □

EDUCATIONAL SOFTWARE

A GUIDE FOR PARENTS
PART V

By Richard Herring

What approach should educational software take to its subject? For better or worse, this is a problem which plagues not only software designers, but all educators. It is the classic conflict between process and product; means and ends. In the schools, emphasis is placed on the product of learning—on whether the student can recite certain facts. Learning in the home tends to be more process-oriented—allowing the child to think through a problem and discover how to solve it.

These two approaches to learning are evident in educational software. Most educational software fits (sometimes uncomfortably) into one of four categories depending on how it presents the learning challenge: drill and practice, tutorial, simulation, and problem solving. The first two, drill and practice and tutorial, rely on teaching facts. These types of programs fit neatly into school curricula; the learning that takes place is practical and measurable. Simulation and problem solving programs, on the other hand, offer the child an environment to explore and promote the development of insight.

Drill and practice programs are fine (and very common) for subjects best-learned through memorization. Offering immediate reinforcement and remediation, many programs are able to present questions randomly so that the child can't just memorize a series of answers. Some software can even individualize the presentation of questions based on the child's progress and previous scores. Drill and practice programs are available for almost all grades and school subjects.

One subject for which the drill and practice approach works well is reading speed. *Speed Reader II,* from Davidson & Associates, is a flexible example. Along with the program disk, a separate data disk contains thirty-five reading selections; additional data disks are available for any grade level. You can start with warm-up exercises, practice eye movement, and take timed reading tests. Other exercises help your peripheral vision and eye span as well as train special skills like reading columns. So that you do not emphasize speed to the detriment of understanding, comprehension tests are also given. If you want to enter your own text, *Speed Reader II* will analyse it and tell you what grade level it's on.

Another approach to reading and word recognition skills is taken in Didatech's *Fay: The Word Hunter.* In this word search game, the heroine, Fay, searches underwater for words hidden in rows of letters and shoots those words with her laser. Of the 3000 words on the disk, those that are frequently misspelled will appear most often. This program uses a more arcadelike approach than a static or printed word-search puzzle.

The trophy for most creative use of the computer's capabilities in the drill and practice category may have to go to *Chatterbee,* a spelling program from Tronix. Spelling programs have used many innovative approaches to avoid showing the word on screen and allowing the child to copy it. *Chatterbee* talks. It actually says each word, then uses it in a sentence. When the child types the word correctly, he gets verbal reinforcement and a smile from the Chatterbee. If the child guesses wrong three times,

the program spells the word out loud. The child can start at any of twelve levels, but the program will automatically adjust to the correct level for the child based on his performance.

Chatterbee violates one rule for slow learners: do not announce the child's failure. If adults are within earshot, the child may hesitate to try, knowing that answers will be announced by the computer. For children who require tutoring for specific language learning disabilities, SLED Software offers a series of spelling programs designed by a language therapist. In addition to drill and practice on spelling rules, SLED's programs use exercises and tutorials.

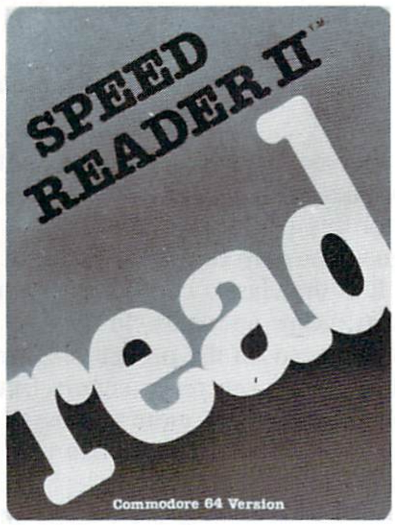
Drill and practice programs are fine at school where kids are a captive audience. But in the home environment, those programs often require parental interest and involvement, or a highly motivated child. At home, there are just so many diversions which aren't available at school. Drill and practice is basically learning through repetition. (Notice I did not mention any math programs; a ton of them exist.) As they begin to include detailed explanations for right and wrong answers, drill and practice programs start to look like tutorials. Programs which prepare high school students for the Scholastic Aptitude Test are great examples.

One reasonably priced program ($30) is *SAT Review* from Moses Engineering. This package of five cassettes includes a practice exam, skill training in English and math, and a final exam. SAT programs are heavily drill and practice oriented, yet programs like *SAT Review* begin to tutor you on the facts needed to do well on the college boards.

Another approach to SAT study is Harcourt Brace Jovanovich's *Computer Preparation/SAT* which comes with 540 drill questions, 1000 vocabulary 'flash cards' and four full length tests. The tests are actually taken in a workbook, with answers typed on the computer. This approach for the tests is just about as boring as using a $9.95 SAT review manual from the local bookstore. The drill questions, however, are really tutorial in nature. In the synonym section, a word will be displayed not only with a list of possible synonyms, but also with hints on how to approach the problem and finally, a definition.

True tutorials first give the child basic information on the subject and then follow up with questions (often multiple choice) on the material. When the child gives an incorrect answer, the program will usually return to the lesson to review or further explain the fact. Subjects like history, geography, and sciences are suitable for tutorial presentation.

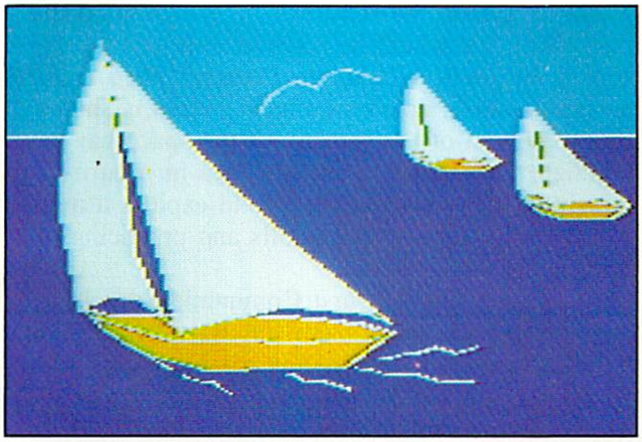
Early Music Games is a program designed to tutor your child on piano keyboard skills. While correct finger positioning cannot be learned on a computer keyboard, the screen displays of a piano keyboard and bass and treble clefs will familiarize your child with important concepts. The top row of keys is used to simulate a keyboard. As your child "types" notes, they are shown on the screen keyboard. (Since half notes—the black keys—are not used, this can serve only as an introduction.) In bass or treble clef mode, the tune your child plays appears as notes on a staff. Once the child begins to grasp all this, he can test his understanding by having the computer display a series of notes from which he must identify the unlabeled one.



Speed Reader II ***uses the drill and practice approach. Program includes warm-up exercises, eye movement practice, and timed reading tests.***
READER SERVICE NO. 5

Some tutorials are even aimed at teaching you how to use computer software. *Logo Locomotion* is a series of games which help children learn turtle graphics. (It must be used with *Commodore Logo.*) Different grids are shown on screen to help the child get a feel for the angles and distances through which the Logo turtle moves. Feedback is given for the commands the child types so that mistakes can be corrected quickly. When the child develops an intuitive feel for distances and relative and absolute headings, he can continue to play the games without the grids to guide him. Since the *Logo Locomotion* programs can be listed, real mastery may come from understanding and modifying the programs themselves.

Tutorials are a good approach when you want your child to learn about a subject generally. He will pick up whatever facts seem interesting and probably ignore the rest, but he will put it all in a new contextual framework of how volcanoes erupt



Delta Drawing ***brings turtle graphics to toddlers.***
READER SERVICE NO. 6

or whatever. Examine how the program will treat the child. As additional correct responses are given, drill-and-practice programs and tutorials should become gradually more difficult. And, if you give a wrong answer, the program should back up a bit or maintain its current level of difficulty, but not penalize you by forcing you to start with the easiest problems.

After drill and practice and tutorials, we move into simulations and problem solving programs. Both generally recognize many correct answers to any question. David Seuss, president of Spinnaker Software, points out that educational software may be oriented toward a specific subject, as are many spelling and arithmetic programs, or that it may provide an environment to explore, like simulation and adventure programs. He asks which software you think your child is most likely to want to use again and again.

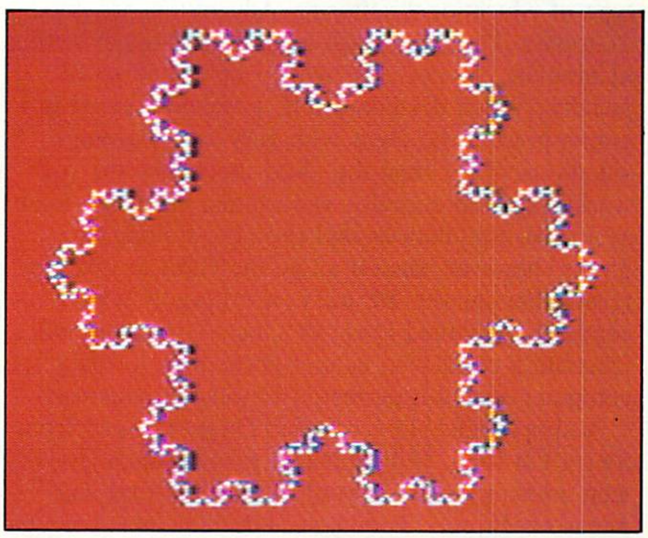
Typically best for teaching about systems, there are numerous simulations that allow you to rule an empire and force you to make hard economic decisions. By providing a feeling for the subject, simulations let your child develop values and understand complex dynamic relationships. In a simulation, the child is often asked a "what if" question. Based on any of a wide variety of answers, the computer will tell your child what difference his response made. Eventually, your child will begin to understand the dynamic relationship among the variables in the simulation; he will learn how parts of systems interrelate.

Sandra Curtis, director of research for Joyce Hakansson Associates, describes how she tries to develop software that will lead to insight, the 'ah-ha experience' as she calls it. Little children are like little scientists, learning through trial and error. Dr. Curtis gives the example of a toddler dropping a spoon again and again at mealtime. This is a great discovery; release the spoon and it falls. Regardless how much it seems that this repetitious behavior is just the child trying to annoy the parent, it is really an 'ah-ha experience.'

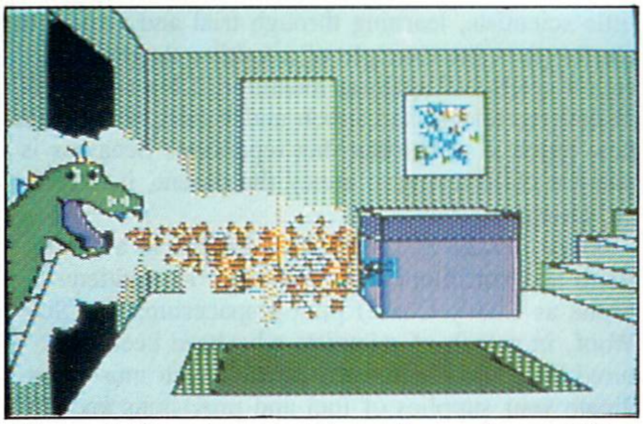
Space Rescue is a simulation hidden in a game. Benji and you (hopefully you're not as accident-prone as Chevy Chase) pilot a spaceship, The Star Woof, in search of scientists who have been captured by aliens. Before the journey, you must coordinate your supplies of fuel and provisions and set your ship's course. Statistics in this program were obtained from NASA; fuel and resources relate directly to distance traveled. You will learn the layout of our solar system and have a few arcade episodes with attacking alien ships and meteor storms. The strength of this program is its gentle introduction to just a few of the intricacies of space flight.

Simulations often use this type of hypothetical environment. The hypothetical consequences of our actions are the only way to experience many situations; most of us will never pilot a spaceship. Or, simulations may be used where the activity is just too dangerous. We have all read press accounts of military simulations which are a learning process for those involved, but which have consequences too terrible to imagine in the real world. Sometimes learning through role playing is the only way.

Historical background may play a part in a simulation. Spinnaker's program *Trains* is set in the Old West and based on the economy of the times. Chugging between refineries, oil wells, and coal



Commodore LOGO ***has seven programmable sprites.***
READER SERVICE NO. 7



Dragon's Keep ***teaches kids reading and map skills.***
READER SERVICE NO. 8

depots, your train will help you learn about the law of supply and demand. As you transport lumber to the mill or ore to the factory with your coal burner, you will have to plan ahead. What are the most lucrative goods to transport? Can you make one more run without running out of coal for the locomotive? To advance to the higher rounds, you will have to get better and better at juggling variables, at exploring the limits of the system.

Simulations like this offer children an exciting medium for developing and honing their analytical thinking skills. Critics sometimes point out that simulations oversimplify reality, that they make children think superficially about a complex world. But isn't that the natural process of learning: to understand pieces of the whole; incomplete at first, but becoming more and more detailed?

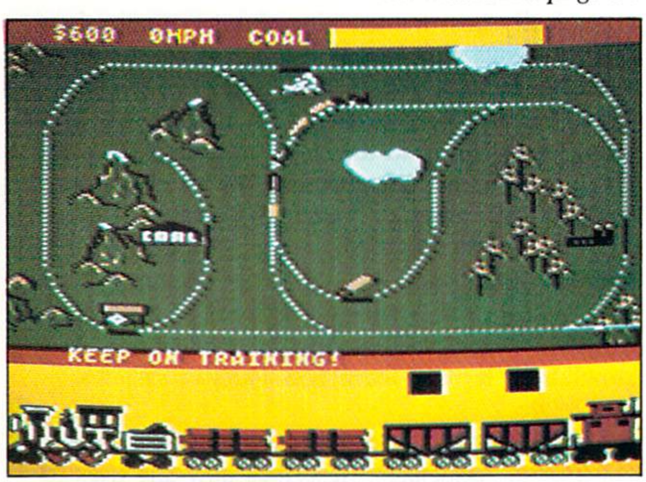
Problem-solving programs are the last general type of educational software. Adventures are good examples. The child must explore a fictional setting, manipulate objects, deal with people, and solve puzzles. When designed well, problem-solving programs present the child with very real learning situations. They treat the child much like the real world does—in complex ways rather than one issue at a time. Simulations and problem-solving programs, however, should allow the child to give any reasonable answer. Be wary of programs which accept a very limited range of inputs since they will frustrate a child who cannot guess how to make the computer accept a perfectly reasonable response.

A simple adventure game is *Dragon's Keep* from Sierra On-Line. With a magic dragon keeping sixteen animals prisoner, your child must find and release each one. Twenty-six different locations can be explored. The program comes with a map and a sticker for each animal. As your child explores the world on screen, he can track his progress on the map and mark where he rescued each animal. Designed for younger children, *Dragon's Keep* requires no typing skills. Each movement option is shown on the bottom of the screen; hit the space bar to highlight the one you want and then hit return. This delightful game will let your child explore a mini-world while learning map skills and practicing reading.

The Factory by Sunburst Communications is another example of problem-solving software. Using lots of graphics and animation, this program will challenge your child to produce certain shapes on a geometric assembly line. First, your child gets to see how the assembling machines work. Next, he will set up a series of machines to make a product, probably building as he goes, with only a vague idea what the finished shape will look like. In the end, he will have to design an assembly line that can produce a shape determined by the program. Like programming, *The Factory* stresses how important sequence is in accomplishing goals. It also teaches inductive reasoning; the child must learn to think backward from the goal through the steps necessary to accomplish it.

The best problem solving software around is actually a computer language—Logo (or Pilot). Using turtle graphics, a child can not only explore the world where the turtle lives, but can also create his own miniature landscape. This is more than simple drawing; it is logic and geometry and motion. There is little the turtle can do that the child cannot do by pacing out a precisely described path. In fact, that is one of the best ways to understand turtle graphics.

Continued on page 98



Simulations like **Trains** ***develop analytical skills.***
READER SERVICE NO. 9

PROGRAM LISTINGS

On the following pages are listed several programs that we hope you'll want to punch in your Commodore computer. But please read the following introduction first; there are a few things you'll need to know.

Certain computer commands are displayed on the monitor by a variety of odd-looking characters. To get your computer to display these commands rather than actually perform them, you'll need to enter the quote mode. Hold down the SHIFT key and press the "2" key; a set of quote marks will appear. This tells the computer that the characters that follow are to be displayed, not performed. To exit the quote mode, type another set of quote marks, or hit the RETURN key. You'll also enter the quote mode when you INserT spaces or characters onto a line.

In *Ahoy!'s* program listings, you'll frequently find letters and/or numbers surrounded by brackets { }. That's because, for the purposes of clear reproduction, we at *Ahoy!* use a daisy wheel printer incapable of reproducing command symbols. For example, when you're in the quote mode and press the SHIFT and CLR/HOME keys at the same time, the screen (or a dot-matrix printer) will indicate this command with a heart {♥}. Because a daisy wheel cannot duplicate this symbol, it substitutes an alternate code between brackets. In the case of the SHIFT/CLR HOME symbol, our printer substitutes {SC}.

Another special case is SHIFT and COMMODORE characters. We represent these by underlining or overlining, respectively: any character underlined in the program listing should be punched in as a SHIFTed character ($\underline{J}$ = SHIFT J), any character overlined should be punched in as a COMMODORE character ($\overline{J}$ = COMMODORE J).

An alternate way of entering commands and other graphics symbols and characters is to use their corresponding character strings. The CLR/HOME command, for example, is entered by typing CHR$(147). While this requires a few extra strokes, it facilitates editing your program or reading the printed listing. For a complete list of CHR$ codes, consult the appendix at the back of your Commodore user manual.

Below is a list of the command abbreviations you'll find in our program listings, the commands they stand for, how to enter them, and how they'll appear on the screen or on a dot matrix printout.

When You See	It Means	You Type	You Will See	When You See	It Means	You Type	You Will See
{SC}	Screen Clear	SHIFT CLR/HOME		{YL}	Yellow	CNTRL 8	
{HM}	Home	CLR/HOME		{OR}	Orange	COMMODORE 1	
{CU}	Cursor Up	SHIFT ↑ CRSR ↓		{BR}	Brown	COMMODORE 2	
{CD}	Cursor Down	↑ CRSR ↓		{LR}	Light Red	COMMODORE 3	
{CL}	Cursor Left	SHIFT ← CRSR →		{G1}	Grey 1	COMMODORE 4	
{CR}	Cursor Right	← CRSR →		{G2}	Grey 2	COMMODORE 5	
{SS}	Shifted Space	SHIFT space		{LG}	Light Green	COMMODORE 6	
{IN}	Insert	INST		{LB}	Light Blue	COMMODORE 7	
{RV}	Reverse On	CNTRL 9		{G3}	Grey 3	COMMODORE 8	
{RO}	Reverse Off	CNTRL 0		{F1}	Function 1	F 1	
{BK}	Black	CNTRL 1		{F2}	Function 2	F 2	
{WH}	White	CNTRL 2		{F3}	Function 3	F 3	
{RD}	Red	CNTRL 3		{F4}	Function 4	F 4	
{CY}	Cyan	CNTRL 4		{F5}	Function 5	F 5	
{PU}	Purple	CNTRL 5		{F6}	Function 6	F 6	
{GN}	Green	CNTRL 6		{F7}	Function 7	F 7	
{BL}	Blue	CNTRL 7		{F8}	Function 8	F 8	

BUG REPELLENT CORRECTION
These are updated versions of *Bug Repellent*. See the note on page 4 before typing any programs!

VIC 20 BUG REPELLENT

By Michael Kleinert and David Barron

The program listed below will allow you to quickly debug any *Ahoy!* program you type in on your VIC 20. Follow directions for cassette or disk.

For cassette: type in and save the *Bug Repellent* program, then type RUN 63000[RETURN]SYS 828[RETURN]. If you typed the program properly, it will generate a set of two-letter line codes that will match those listed below the program on this page. (If you didn't type the program properly, of course, no line codes will be generated. You'll have to debug the *Bug Repellent* itself the hard way.)

Once you've got a working *Bug Repellent*, type in the program you wish to check. Save it and type the RUN and SYS commands listed above once again, then compare the line codes generated to those listed in the magazine. If you spot a discrepancy, a typing error exists in that line. Important: you must use exactly the same spacing as the program in the magazine. Due to memory limitations on the VIC, the VIC *Bug Repellent* will register an error if your spacing varies from what's printed.

You may type SYS 828 as many times as you wish, but if you use the cassette for anything, type RUN 63000 to restore the *Repellent*.

When your program has been disinfected you may delete all lines from 63000 on. (Be sure the program you type doesn't include lines above 63000!)

For disk: type in the *Bug Repellent*, save it, and type RUN:NEW[RETURN]. (See above regarding testing the *Bug Repellent* on itself.) Type in the program you wish to check, then SYS 828. This will generate a set of two-letter line codes that you should compare to those listed in the magazine.

To pause the line codes listing, press SHIFT. To permanently pause it, press SHIFT LOCK. To continue, release SHIFT LOCK.

To send the list to the printer type OPEN 4,4:CMD 4:SYS 828[RETURN]. When the cursor comes back, type PRINT#4:CLOSE 4[RETURN].

```
•63000 FOR X = 828 TO 1023 :READ Y
:POKE X,Y:NEXT:END
•63001 DATA 169, 0, 133, 63, 133,
64, 165, 43, 133, 251
•63002 DATA 165, 44, 133, 252, 160
, 0, 132, 254, 32, 228
•63003 DATA 3, 234, 177, 251, 208,
 3, 76, 208, 3, 230
•63004 DATA 251, 208, 2, 230, 252,
 169, 244, 160, 3, 32
•63005 DATA 30, 203, 160, 0, 177,
251, 170, 230, 251, 208
•63006 DATA 2, 230, 252, 177, 251,
 32, 205, 221, 169, 58
•63007 DATA 32, 210, 255, 169, 0,
133, 253, 230, 254, 32
•63008 DATA 228, 3, 234, 165, 253,
 160, 0, 170, 177, 251
•63009 DATA 201, 32, 240, 6, 138,
113, 251, 69, 254, 170
•63010 DATA 138, 133, 253, 177, 25
1, 208, 226, 165, 253, 41
•63011 DATA 240, 74, 74, 74, 74, 2
4, 105, 65, 32, 210
•63012 DATA 255, 165, 253, 41, 15,
 24, 105, 65, 32, 210
•63013 DATA 255, 169, 13, 32, 210,
 255, 173, 141, 2, 41
•63014 DATA 1, 208, 249, 230, 63,
208, 2, 230, 64, 230
•63015 DATA 251, 208, 2, 230, 252,
 76, 74, 3, 169, 236
•63016 DATA 160, 3, 32, 30, 203, 1
66, 63, 165, 64, 32
•63017 DATA 205, 221, 169, 13, 32,
 210, 255, 96, 230, 251
•63018 DATA 208, 2, 230, 252, 96,
0, 76, 73, 78, 69
•63019 DATA 83, 58, 32, 0, 76, 73,
 78, 69, 32, 35
•63020 DATA 32, 0, 0, 0, 0, 0
```

BUG REPELLENT LINE CODES FOR VIC 20 BUG REPELLENT

```
LINE # 63000:MH    LINE # 63011:NN
LINE # 63001:BD    LINE # 63012:IG
LINE # 63002:FO    LINE # 63013:EN
LINE # 63003:ND    LINE # 63014:GJ
LINE # 63004:DJ    LINE # 63015:IK
LINE # 63005:LP    LINE # 63016:HG
LINE # 63006:JB    LINE # 63017:CK
LINE # 63007:JF    LINE # 63018:JF
LINE # 63008:KA    LINE # 63019:OH
LINE # 63009:HP    LINE # 63020:LH
LINE # 63010:KJ    LINES: 21
```

C-64 BUG REPELLENT

By Michael Kleinert and David Barron

The program listed below will allow you to quickly debug any *Ahoy!* program you type in on your C-64.

Type in and save the *Bug Repellent* program. Type NEW, then type in the *Ahoy!* program you wish to check. When that's done, save your program (don't run it!) and type SYS 49152 RETURN. You'll be asked if you want the line value codes displayed on the screen or dumped to the printer. If you select screen, it will appear there.

The table will move quickly, too quickly for most mortals to follow. To pause the listing depress and hold the SHIFT key. To pause for an extended period, depress SHIFT LOCK. As long as it is locked, the display will remain frozen.

Compare the table your machine generates to the table in *Ahoy!* that follows the program you're entering. If you spot a difference, an error exists in that line. Jot down the numbers of lines where contradictions occur, LIST each line, spot the errors, and correct them.

```
•5000 FOR X = 49152 TO 49488 :READ
 Y:POKE X,Y:NEXT:END
•5001 DATA 32, 161, 192, 165, 43,
133, 251, 165, 44, 133
•5002 DATA 252, 160, 0, 132, 254,
32, 37, 193, 234, 177
•5003 DATA 251, 208, 3, 76, 138, 1
92, 230, 251, 208, 2
•5004 DATA 230, 252, 76, 43, 192,
76, 73, 78, 69, 32
•5005 DATA 35, 32, 0, 169, 35, 160
, 192, 32, 30, 171
•5006 DATA 160, 0, 177, 251, 170,
230, 251, 208, 2, 230
•5007 DATA 252, 177, 251, 32, 205,
 189, 169, 58, 32, 210
•5008 DATA 255, 169, 0, 133, 253,
230, 254, 32, 37, 193
•5009 DATA 234, 165, 253, 160, 0,
76, 13, 193, 133, 253
•5010 DATA 177, 251, 208, 237, 165
, 253, 41, 240, 74, 74
•5011 DATA 74, 74, 24, 105, 65, 32
, 210, 255, 165, 253
•5012 DATA 41, 15, 24, 105, 65, 32
, 210, 255, 169, 13
•5013 DATA 32, 220, 192, 230, 63,
208, 2, 230, 64, 230
•5014 DATA 251, 208, 2, 230, 252,
76, 11, 192, 169, 153
•5015 DATA 160, 192, 32, 30, 171,
166, 63, 165, 64, 76
•5016 DATA 231, 192, 96, 76, 73, 7
8, 69, 83, 58, 32
•5017 DATA 0, 169, 247, 160, 192,
32, 30, 171, 169, 3
•5018 DATA 133, 254, 32, 228, 255,
 201, 83, 240, 6, 201
•5019 DATA 80, 208, 245, 230, 254,
 32, 210, 255, 169, 4
•5020 DATA 166, 254, 160, 255, 32,
 186, 255, 169, 0, 133
•5021 DATA 63, 133, 64, 133, 2, 32
, 189, 255, 32, 192
•5022 DATA 255, 166, 254, 32, 201,
 255, 76, 73, 193, 96
•5023 DATA 32, 210, 255, 173, 141,
 2, 41, 1, 208, 249
•5024 DATA 96, 32, 205, 189, 169,
13, 32, 210, 255, 32
•5025 DATA 204, 255, 169, 4, 76, 1
95, 255, 147, 83, 67
•5026 DATA 82, 69, 69, 78, 32, 79,
 82, 32, 80, 82
•5027 DATA 73, 78, 84, 69, 82, 32,
 63, 32, 0, 76
•5028 DATA 44, 193, 234, 177, 251,
 201, 32, 240, 6, 138
•5029 DATA 113, 251, 69, 254, 170,
 138, 76, 88, 192, 0
•5030 DATA 0, 0, 0, 230, 251, 208,
 2, 230, 252, 96
•5031 DATA 170, 177, 251, 201, 34,
 208, 6, 165, 2, 73
•5032 DATA 255, 133, 2, 165, 2, 20
8, 218, 177, 251, 201
•5033 DATA 32, 208, 212, 198, 254,
 76, 29, 193, 0, 169
•5034 DATA 13, 76, 210, 255, 0, 0,
 0
```

BUG REPELLENT LINE CODES FOR C-64 BUG REPELLENT

```
LINE # 5000:GJ       LINE # 5018:FK
LINE # 5001:DL       LINE # 5019:FL
LINE # 5002:DB       LINE # 5020:CL
LINE # 5003:OF       LINE # 5021:GC
LINE # 5004:KN       LINE # 5022:NN
LINE # 5005:CA       LINE # 5023:NH
LINE # 5006:CE       LINE # 5024:IM
LINE # 5007:JE       LINE # 5025:KC
LINE # 5008:CL       LINE # 5026:DC
LINE # 5009:NB       LINE # 5027:ML
LINE # 5010:MB       LINE # 5028:GN
LINE # 5011:EP       LINE # 5029:JK
LINE # 5012:GH       LINE # 5030:NA
LINE # 5013:AN       LINE # 5031:DM
LINE # 5014:NG       LINE # 5032:JA
LINE # 5015:BF       LINE # 5033:FM
LINE # 5016:EP       LINE # 5034:PA
LINE # 5017:PJ       LINES: 35
```

WHAT'S MY JOB?

FROM PAGE 23

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```
10 POKE53280,11:POKE53281,0:PRINTCHR$(14)
20 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}";TAB(12);"{CD}{G3}{RV} WHAT'S MY JOB? {RO}"
30 PRINTSPC(9);"{CD}{CD}{LB}A PROGRAM FOR THE C-64"
40 PRINTSPC(13);"{CD}{CD}BY B.W.BEHLING"
50 FORI=0TO3000:NEXTI
60 DIMA$(2000):FORI=0TO3:READA$(I):NEXTI
70 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD} DO YOU WANT ALL PREVIOUS JOBS ENTERED?"
75 GETK$:IFK$=""THEN75
80 IFK$<>"Y"THEN100
85 PRINTSPC(3);"{CD}{CD}{CD}{CD}{CD}{CD}HIT [*] WHEN 'JOBS' TAPE IS READY."
90 GETK$:IFK$=""THEN90
95 IFK$="*"THENGOSUB2000
100 PRINT"{SC}";TAB(12);"{CD}{G3}{RV} WHAT'S MY JOB? {RO}"
110 PRINT"{CD}{CD}{LG} YOU THINK OF A PERSON AND I'LL TRY TO"
120 PRINT" GUESS WHAT KIND OF WORK THEY DO."
122 PRINT"{CD} IF THE ANSWER TO A QUESTION IS NOT YES"
124 PRINT" MORE THAN 50% OF THE TIME ANSWER NO."
126 PRINTSPC(7);"{CD}{CD}{OR}HIT [L] TO LIST ALL JOBS."
130 PRINT"{CD}{CD}{LG} ARE YOU READY?"
135 GETA$:IFA$=""THEN135
140 IFA$="L"ORA$="L"THEN600
150 IFA$="N"ORA$="N"THEN700
160 K=1
170 GOSUB390
180 IFLEN(A$(K))=0THEN700
190 IFLEFT$(A$(K),2)="/Q"THEN170
200 PRINT"{RV}{CD}ARE YOU A ";RIGHT$(A$(K),LEN(A$(K))-2);
210 INPUTA$
220 A$=LEFT$(A$,1)
225 IFA$<>"Y"THEN240
230 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{LG}       {RV} TRY ANOTHER PERSON [Y/N]? "
234 GETK$:IFK$=""THEN234
236 IFK$="Y"ORK$="Y"THEN100
238 GOTO700
240 PRINT "{CD}{LG}I GIVE UP!!  WHAT IS YOUR JOB?"
242 PRINT"{CD}{OR} [LIMIT{SS}19{SS}LETTERS AND{SS}SPACES]{YL}{CD}"
244 INPUTV$
246 IFLEN(V$)>19THENPRINTSPC(4);"{CD}{LB} ERROR!{SS}JOB{SS}TOO{SS}LONG!!":GOTO242
250 PRINT"{CD}{LG} PLEASE GIVE ME A YES OR NO QUESTION TO TELL A ";
260 PRINTV$;" FROM A ";RIGHT$(A$(K),LEN(A$(K))-2);"?{CD}"
265 PRINT"{OR} [LIMIT{SS}38{SS}LETTERS AND{SS}SPACES]"
270 INPUT"{CD}{YL}";X$
275 IFLEN(X$)>38THENPRINTSPC(3);"{CD}{LB}ERROR!{SS}QUESTION{SS}TOO{SS}LONG!!{CD}":GOTO265
280 PRINT"{LG}FOR A ";V$;" THE ANSWER WOULD BE?"
290 INPUT"{YL}";A$
300 A$=LEFT$(A$,1):IFA$<>"Y"ANDA$<>"N"THEN280
310 IFA$="Y"THENB$="N"
320 IFA$="N"THENB$="Y"
330 Z1=VAL(A$(0)):A$(0)=STR$(Z1+2):A$(Z1)=A$(K):A$(Z1+1)="/A"+V$
370 A$(K)="/Q"+X$+"/"+A$+STR$(Z1+1)+"/"+B$+STR$(Z1)+"/"
380 GOTO100
390 REM   SUBROUTINE TO PRINT QUESTIONS
400 Q$=A$(K)
410 FORZ=3TOLEN(Q$)
415 IFMID$(Q$,Z,1)<>"/"THENPRINTMID$(Q$,Z,1);:NEXTZ
420 INPUTC$
430 C$=LEFT$(C$,1)
440 IFC$<>"Y"ANDC$<>"N"THEN410
450 G$="/"+C$
```

```
•455 FORX=3TOLEN(Q$)-1
•460 IFMID$(Q$,X,2)=G$THEN480
•470 NEXTX
•480 FORY=X+1TOLEN(Q$)
•490 IFMID$(Q$,Y,1)="/"THEN510
•500 NEXTY
•510 K=VAL(MID$(Q$,X+2,Y-X-2))
•520 RETURN
•530 DATA"4","/QDO YOU WORK MOSTLY
 OUTDOORS/Y2/N3","/ATRUCK{SS}DRIV
ER","/ATEACHER"
•600 PRINT"{CD}{CD}{LG}JOBS I ALRE
ADY KNOW ARE:{CD}{YL}"
•605 X=0
•610 FORI=1TOVAL(A$(0))
•620 IFLEFT$(A$(I),2)<>"/A"THEN660
•622 IFX=1THENPRINTTAB(20);
•624 IFX=0THENPRINT
•630 FORZ=3TOLEN(A$(I))
•640 IFMID$(A$(I),Z,1)<>"/"THENPRI
NTMID$(A$(I),Z,1);:NEXTZ
•650 X=X+1:IFX>1THENX=0
•660 NEXTI
•662 PRINT:PRINTSPC(8);"{CD}{CD}HI
T ANY KEY TO CONTINUE."
•664 GETK$:IFK$=""THEN664
•670 GOTO100
•700 PRINT"{SC}{CD}{CD}{CD}{CD}{CD
}{CD}  DO YOU WANT TO SAVE THE JO
BS I KNOW?"
•710 GETK$:IFK$=""THEN710
•720 IFK$<>"Y"THEN800
•725 PRINTSPC(3);"{CD}{CD}{CD}{CD}
{CD}{CD}HIT [*] WHEN 'JOBS' TAPE
IS READY."
•730 GETK$:IFK$=""THEN730
•735 IFK$="*"THENGOSUB1000
•800 PRINT"{SC}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{LG}"SPC(16)
;"GOODBYE!":FORI=0TO1000:NEXTI
•810 PRINT"{SC}":END
•1000 OPEN1,1,2,"JOBS A$"
•1005 R$=CHR$(13)
•1010 FORI=0TOVAL(A$(0)):PRINT#1,A
$(I)R$:NEXTI
•1015 CLOSE1:RETURN
•2000 OPEN1,1,0,"JOBS A$"
•2005 FORI=0TO2000
•2010 IFI=VAL(A$(0))THEN2025
•2015 INPUT#1,A$(I)
•2020 NEXTI
•2025 CLOSE1:RETURN
```

BUG REPELLENT LINE CODES FOR WHATS MY JOB?

```
LINE # 10:CL      LINE # 380:CF
LINE # 20:AH      LINE # 390:AB
LINE # 30:AI      LINE # 400:DD
LINE # 40:DG      LINE # 410:PJ
LINE # 50:DB      LINE # 415:KD
LINE # 60:EO      LINE # 420:OJ
LINE # 70:GJ      LINE # 430:AC
LINE # 75:DL      LINE # 440:FC
LINE # 80:CE      LINE # 450:KF
LINE # 85:IF      LINE # 455:MM
LINE # 90:EO      LINE # 460:FJ
LINE # 95:BA      LINE # 470:NK
LINE # 100:DK     LINE # 480:OH
LINE # 110:MB     LINE # 490:KF
LINE # 120:AH     LINE # 500:NN
LINE # 122:AH     LINE # 510:EL
LINE # 124:EK     LINE # 520:IM
LINE # 126:DK     LINE # 530:AP
LINE # 130:BK     LINE # 600:ND
LINE # 135:HO     LINE # 605:DN
LINE # 140:DK     LINE # 610:HF
LINE # 150:ED     LINE # 620:JG
LINE # 160:DD     LINE # 622:JC
LINE # 170:CI     LINE # 624:AH
LINE # 180:PP     LINE # 630:JH
LINE # 190:MB     LINE # 640:PM
LINE # 200:HN     LINE # 650:BL
LINE # 210:PH     LINE # 660:MN
LINE # 220:PO     LINE # 662:PP
LINE # 225:EP     LINE # 664:GL
LINE # 230:GI     LINE # 670:CF
LINE # 234:GI     LINE # 700:OP
LINE # 236:IN     LINE # 710:GL
LINE # 238:CD     LINE # 720:CL
LINE # 240:IO     LINE # 725:IF
LINE # 242:JO     LINE # 730:GN
LINE # 244:AG     LINE # 735:BH
LINE # 246:NP     LINE # 800:CM
LINE # 250:DB     LINE # 810:CI
LINE # 260:LL     LINE # 1000:DK
LINE # 265:PL     LINE # 1005:KN
LINE # 270:EN     LINE # 1010:BG
LINE # 275:AJ     LINE # 1015:JG
LINE # 280:EG     LINE # 2000:CI
LINE # 290:PA     LINE # 2005:DK
LINE # 300:GF     LINE # 2010:OL
LINE # 310:KM     LINE # 2015:OJ
LINE # 320:JG     LINE # 2020:MN
LINE # 330:MG     LINE # 2025:JG
LINE # 370:MC     LINES: 99
```

MATH DEFENDER

FROM PAGE 47

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

VIC 20 VERSION

```
•1 REM MATH DEFENDER
•5 POKE36879,238:GOSUB2000:GOSUB2160
•10 X=RND(-TI):DIMT%(4),X%(3),Y%(3),A%(3),O$(3),P%(3),OP$(4),OX$(4)
•13 S=36876:POKES+2,15:FORN=1TO3:P%(N)=1:NEXT
•14 D$="{HM}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}":SP$="                      "
•15 O$(1)="":O$(2)="{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}":O$(3)=O$(2)+"{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}"
•17 OX$(1)="+":OX$(2)="-":OX$(3)="X":OX$(4)="/"
•30 GOTO102
•32 PRINTLEFT$(D$,P%(Z%))O$(Z%)"{BK}"X%(Z%)"{CL}"Y%(Z%)"{CL}{CL}{CL}"OP$(Z%):RETURN
•33 Z%=RND(1)*3+1:PRINTLEFT$(D$,P%(Z%))O$(Z%)"      ":P%(Z%)=P%(Z%)+1:GOSUB32:RETURN
•34 IFFLTHENRETURN
•35 IFA$="F"THENM%=M%-1-(M%<2):GOTO37
•36 M%=M%+1+(M%>2)
•37 PRINTD$SP$D$O$(M%)"  {RV}{BK}^ ":RETURN
•38 FORP=1TO2E3:NEXT:RETURN
•39 PRINTLEFT$(D$,P%(M%))O$(M%)Q$:RETURN
•40 REM GEN PROBLEM
•41 Z=INT(RND(1)*4)+1:IFT%(Z)=0THEN41
•42 X=INT(RND(1)*10):Y=INT(RND(1)*10):ONZGOSUB43,44,46,47:RETURN
•43 A=X+Y:RETURN
•44 D=X+Y:X=D:A=X-Y:RETURN
•46 A=X*Y:RETURN
•47 IFY=0THENY=INT(RND(1)*10):GOTO47
•48 D=X*Y:X=D:A=X/Y:RETURN
•50 REM GET ANSWER
•51 FL=1:AN$=AN$+A$
•52 PRINTD$"{CD}"SP$D$O$(M%)"{CD}{BK}"AN$:IFLEN(AN$)>2THENGOSUB420
•53 RETURN
•60 PRINTD$"{CD}{CD}{CD}"SP$D$"{CD}{CD}{CD}{RD}SCORE:"SC;:RETURN
•70 IFVAL(AN$)=A%(M%)THENGOSUB400:RETURN
•71 GOSUB420:RETURN
•100 REM INIT
•102 GOSUB1000:GOSUB1500:FORN=1TO3:GOSUB40:OP$(N)=OX$(Z)
•104 X%(N)=X:Y%(N)=Y:A%(N)=A:NEXT:M%=1:GOSUB34
•105 PRINTD$"{CU}{RD}££££££££££££££££££££££":FORN=1TO3:Z%=N:GOSUB32:NEXT
•106 FORP=1TOSK:GETA$:IFA$=""THEN120
•107 IFA$<"0"ORA$>"9"THEN109
•108 GOSUB51:GOTO120
•109 IFA$="F"ORA$="J"THENGOSUB34:GOTO120
•110 IFA$=CHR$(13)ANDLEN(AN$)>0THENGOSUB70
•120 NEXT:GOSUB33:IFP%(Z%)=18THEN502
•122 GOTO106
•400 REM RIGHT ANS
•402 PRINTD$"{CD}"SP$:Q$="{RV}{RD}*****":GOSUB39
•403 FORP=240TO150STEP-4:POKES+1,P:NEXT:POKES+1,0:Q$="      ":GOSUB39
•404 P%(M%)=1:GOSUB41:X%(M%)=X:Y%(M%)=Y:A%(M%)=A:Z%=M%:GOSUB32:AN$="":FL=0:
•405 OP$(M%)=OX$(Z):SC=SC+10:GOSUB60:RETURN
•420 REM WRONG
•422 PRINTD$"{CD}"SP$:FORP=1TO50:POKES-1,130:NEXT:POKES-1,0
•424 AN$="":FL=0:SC=SC-5-5*(SC<5):GOSUB60:RETURN
•500 REM LOSE
•502 FORP=240TO130STEP-.5:POKES,P:NEXT:POKES,0
•504 PRINT"{HM}{CD}  YOUR BASE HAS BEEN        OVERRUN!!!":GOSUB38
•505 PRINT"{HM}"SP$SP$SP$"{HM}{CD}
```

```
 PLAY AGAIN? (Y/N)"
506 GETA$:IFA$="Y"THENRUN10
507 IFA$<>"N"THEN506
510 PRINT"{SC}":END
1000 REM PICK TYPES
1002 PRINT"{SC}CHOOSE ONE OR MORE
      TYPES: (Y/N)"
1004 PRINT"{CD}1. ADDITION";:N=1:
GOSUB1020
1005 PRINT"{CD}2. SUBTRACTION";:N
=2:GOSUB1020
1006 PRINT"{CD}3. MULTIPLICATION"
;:N=3:GOSUB1020
1007 PRINT"{CD}4. DIVISION";:N=4:
GOSUB1022
1008 CO=0:FORN=1TO4:IFT%(N)=1THEN
CO=CO+1
1009 NEXT:IFCO=0THEN1002
1010 PRINT"{SC}":RETURN
1020 REM GET Y/N
1022 GETA$:IFA$<>"Y"ANDA$<>"N"THE
N1022
1024 PRINTTAB(18)A$:IFA$="Y"THENT
%(N)=1:OP$(N)=Q$
1025 RETURN
1500 PRINT"{SC}{CD}{CD}{CD}SKILL
LEVEL?"
1502 PRINT"{CD}{CD}1. BEGINNER":P
RINT"{CD}{CD}2. INTERMEDIATE":PRI
NT"{CD}{CD}3. EINSTEIN{CD}{CD}"
1503 INPUTS$:IFS$>"3"ORS$<"1"THEN
1500
1504 SK=10+(20*(3-VAL(S$))):PRINT
"{SC}":RETURN
2000 PRINT"{SC}"SPC(225)"{PU}MATH
 DEFENDER"SPC(95)"{BL}BY ANDY VAN
 DUYNE"
2003 GOSUB38:PRINT"{BK}{CD}{CD}{C
D}TOUCH {RV}RETURN{RO}..."
2004 GETA$:IFA$<>CHR$(13)THEN2004
2006 RETURN
2100 PRINT"{SC}{CD}{CD}INSTRUCTIO
NS?{SS}(Y/N)"
2102 GETA$:IFA$="N"THENRETURN
2103 IFA$<>"Y"THEN2102
2104 PRINT"{SC}{CD}MOVE THE MARKE
R TO      TARGET A PROBLEM."
2105 PRINT"{CD}{CD}{RV}F{RO} = LE
FT  {RV}J{RO} = RIGHT"
2106 PRINT"{CD}{CD}ANSWER THE PRO
BLEM AND PRESS {RV}RETURN{RO}."
2107 PRINT"{CD}{CD}IF THE PROBLEM
 REACHES THE BOTTOM, THE GAME  IS
 OVER!":GOSUB2003:RETURN
```

BUG REPELLENT LINE CODES FOR VIC 20 MATH DEFENDER

LINE # 1:OC	LINE # 402:AO
LINE # 5:AK	LINE # 403:FK
LINE # 10:NO	LINE # 404:AJ
LINE # 13:HO	LINE # 405:OG
LINE # 14:AK	LINE # 420:CK
LINE # 15:IL	LINE # 422:BA
LINE # 17:JC	LINE # 424:KD
LINE # 30:CH	LINE # 500:LP
LINE # 32:KP	LINE # 502:EK
LINE # 33:GM	LINE # 504:BD
LINE # 34:FI	LINE # 505:LL
LINE # 35:NJ	LINE # 506:DM
LINE # 36:MP	LINE # 507:CB
LINE # 37:LB	LINE # 510:CI
LINE # 38:IB	LINE # 1000:EJ
LINE # 39:AC	LINE # 1002:BF
LINE # 40:HM	LINE # 1004:PP
LINE # 41:NL	LINE # 1005:FH
LINE # 42:NF	LINE # 1006:KE
LINE # 43:BF	LINE # 1007:BA
LINE # 44:EC	LINE # 1008:OH
LINE # 46:CP	LINE # 1009:CO
LINE # 47:FG	LINE # 1010:DG
LINE # 48:FG	LINE # 1020:FL
LINE # 50:GA	LINE # 1022:CF
LINE # 51:PP	LINE # 1024:CJ
LINE # 52:IA	LINE # 1025:IM
LINE # 53:IM	LINE # 1500:ME
LINE # 60:OA	LINE # 1502:ON
LINE # 70:NJ	LINE # 1503:BB
LINE # 71:PP	LINE # 1504:NN
LINE # 100:LM	LINE # 2000:NF
LINE # 102:MM	LINE # 2003:BJ
LINE # 104:NF	LINE # 2004:FO
LINE # 105:BK	LINE # 2006:IM
LINE # 106:OK	LINE # 2100:GJ
LINE # 107:KP	LINE # 2102:KF
LINE # 108:FO	LINE # 2103:HL
LINE # 109:BB	LINE # 2104:FF
LINE # 110:IK	LINE # 2105:JM
LINE # 120:CD	LINE # 2106:IN
LINE # 122:CL	LINE # 2107:AD
LINE # 400:PK	LINES: 85

C-64 VERSION

```
1 REM MATH DEFENDER
2 GOSUB10000
5 POKE53280,0:POKE53281,0:GOSUB21
00
```

```
10 X=RND(-TI):DIMT%(4),X%(5),Y%(5
),A%(5),O$(5),P%(5),OP$(5),OX$(4)
12 S=54272:FORN=STO54295:POKEN,0:
NEXT:POKEN,15:POKES+5,15:POKES+6,
255
13 FORN=1TO5:P%(N)=1:NEXT
14 D$="{HM}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}":SP$="

"
15 O$(1)="":O$(2)="{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}":O$(3)=O$(2)+"
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}"
16 O$(4)=O$(3)+"{CR}{CR}{CR}{CR}{
CR}{CR}{CR}{CR}":O$(5)=O$(4)+"{CR
}{CR}{CR}{CR}{CR}{CR}{CR}{CR}"
17 OX$(1)="+":OX$(2)="-":OX$(3)="
X":OX$(4)="/"
30 GOTO102
32 PRINTLEFT$(D$,P%(Z%))O$(Z%)"{Y
L}"X%(Z%)"{CL}"Y%(Z%)"{CL}{CL}{CL
}"OP$(Z%):RETURN
33 Z%=RND(1)*5+1:PRINTLEFT$(D$,P%
(Z%))O$(Z%)"      ":P%(Z%)=P%(Z%)+
1:GOSUB32:RETURN
34 IFFLTHENRETURN
35 IFA$="F"THENM%=M%-1-(M%<2):GOT
O37
36 M%=M%+1+(M%>4)
37 PRINTD$SP$D$O$(M%)"  {RV}{LG}^
":RETURN
38 FORP=1TO2E3:NEXT:RETURN
39 PRINTLEFT$(D$,P%(M%))O$(M%)Q$:
RETURN
40 REM GEN PROBLEM
41 Z=INT(RND(1)*4)+1:IFT%(Z)=0THE
N41
42 X=INT(RND(1)*10):Y=INT(RND(1)*
10):ONZGOSUB43,44,46,47:RETURN
43 A=X+Y:RETURN
44 D=X+Y:X=D:A=X-Y:RETURN
46 A=X*Y:RETURN
47 IFY=0THENY=INT(RND(1)*10):GOTO
47
48 D=X*Y:X=D:A=X/Y:RETURN
50 REM GET ANSWER
51 FL=1:AN$=AN$+A$
52 PRINTD$"{CD}"SP$D$O$(M%)"{CD}
{PU}"AN$:IFLEN(AN$)>2THENGOSUB420
53 RETURN
60 PRINTD$"{CD}{CD}{CD}"SP$D$"{CD
}{CD}{CD}{LB}{RV}SCORE:"SC:RETURN
70 IFVAL(AN$)=A%(M%)THENGOSUB400:
RETURN
71 GOSUB420:RETURN
100 REM INIT
102 GOSUB1000:GOSUB1500:FORN=1TO5
:GOSUB40:OP$(N)=OX$(Z)
103 X%(N)=X:Y%(N)=Y:A%(N)=A:NEXT:
M%=1:GOSUB34
104 PRINTD$"{CU}{RD}£££££££££££££
£££££££££££££££££££££££££"
105 FORN=1TO5:Z%=N:GOSUB32:NEXT
106 FORP=1TOSK:GETA$:IFA$=""THEN1
20
107 IFA$<"0"ORA$>"9"THEN109
108 GOSUB51:GOTO120
109 IFA$="F"ORA$="J"THENGOSUB34:G
OTO120
110 IFA$=CHR$(13)ANDLEN(AN$)>0THE
NGOSUB70
120 NEXT:GOSUB33:IFP%(Z%)=19THEN5
01
122 GOTO106
400 REM RIGHT ANS
402 PRINTD$"{CD}"SP$:Q$="{RV}{RD}
*****":GOSUB39:POKES+4,129:FORP=1
00TO20STEP-4
403 POKES+1,P:NEXT:POKES+1,0:POKE
S+4,128:Q$="      ":GOSUB39
404 P%(M%)=1:GOSUB41:X%(M%)=X:Y%(
M%)=Y:A%(M%)=A:Z%=M%:GOSUB32:AN$=
"":FL=0
405 OP$(M%)=OX$(Z):SC=SC+10:GOSUB
60:RETURN
420 REM WRONG
422 PRINTD$"{CD}"SP$:POKES+4,33:F
ORP=1TO50:POKES+1,10:NEXT:POKES+1
,0:POKES+4,32
424 AN$="":FL=0:SC=SC-5-5*(SC<5):
GOSUB60:RETURN
500 REM LOSE
501 POKES+4,33
502 FORP=100TO10STEP-.5:POKES+1,P
:NEXT:POKES+1,0:POKES+4,32
504 PRINT"{HM}{CD}    YOUR BASE HA
S BEEN OVERRUN!!!":GOSUB38
505 PRINT"{HM}"SP$SP$SP$"{HM}{CD}
PLAY AGAIN? (Y/N)"
506 GETA$:IFA$="Y"THENRUN10
507 IFA$<>"N"THEN506
510 PRINT"{SC}":END
1000 REM PICK TYPES
1002 PRINT"{SC}{LB}CHOOSE ONE OR
MORE TYPES: (Y/N)"
```

```
1004 PRINT"{CD}1. ADDITION";:N=1:GOSUB1020
1005 PRINT"{CD}2. SUBTRACTION";:N=2:GOSUB1020
1006 PRINT"{CD}3. MULTIPLICATION";:N=3:GOSUB1020
1007 PRINT"{CD}4. DIVISION";:N=4:GOSUB1022
1008 CO=0:FORN=1TO4:IFT%(N)=1THENCO=CO+1
1009 NEXT:IFCO=0THEN1002
1010 PRINT"{SC}":RETURN
1020 REM GET Y/N
1022 GETA$:IFA$<>"Y"ANDA$<>"N"THEN1022
1024 PRINTTAB(18)A$:IFA$="Y"THENT%(N)=1:OP$(N)=Q$
1025 RETURN
1500 PRINT"{SC}{CD}{CD}{CD}SKILL LEVEL?"
1502 PRINT"{CD}{CD}1. BEGINNER":PRINT"{CD}{CD}2. INTERMEDIATE":PRINT"{CD}{CD}3. EINSTEIN{CD}{CD}"
1503 INPUTS$:IFS$>"3"ORS$<"1"THEN1500
1504 SK=5+(10*(3-VAL(S$))):PRINT"{SC}":RETURN
2003 PRINT"{WH}{CD}{CD}{CD}TOUCH {RV}RETURN{RO}..."
2004 GETA$:IFA$<>CHR$(13)THEN2004
2006 RETURN
2100 PRINT"{SC}{CD}{CD}{GN}INSTRUCTIONS?{SS}(Y/N)"
2102 GETA$:IFA$="N"THENRETURN
2103 IFA$<>"Y"THEN2102
2104 PRINT"{SC}{CD}MOVE THE MARKER TO TARGET A PROBLEM."
2105 PRINT"{CD}{CD}{RV}F{RO} = LEFT  {RV}J{RO} = RIGHT"
2106 PRINT"{CD}{CD}ANSWER THE PROBLEM AND PRESS {RV}RETURN{RO}."
2107 PRINT"{CD}{CD}IF A PROBLEM REACHES THE BOTTOM, THE      GAME IS OVER!"
2108 GOSUB2003:RETURN
10000 REM TITLE PAGE
10001 POKE53280,0:POKE53281,0
10002 PRINT"{SC}{CD}{CD}{RD}*************************"
10003 FORN=1TO5
10004 PRINT"*                       *":NEXT
10005 PRINT"*************************"
10006 PRINT"{HM}{CD}{CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{YL}";
10007 PRINT"{RV}                       "
10008 PRINT"{CR}{CR}{CR}{CR}{CR}{CR}{CR}{RV}                       "
10009 PRINT"{CR}{CR}{CR}{CR}{CR}{CR}{CR}{RV}   **MATH DEFENDER**   "
10010 PRINT"{CR}{CR}{CR}{CR}{CR}{CR}{CR}{RV}                       "
10011 PRINT"{CR}{CR}{CR}{CR}{CR}{CR}{CR}{RV}                       "
10013 GOSUB38
10019 GOSUB38
10020 PRINT"{CD}{CD}{CD}{CD}{LB}          BY ANDY VAN DUYNE" :GOSUB2003:PRINT"{SC}":RETURN
```

BUG REPELLENT LINE CODES FOR C-64 MATH DEFENDER

LINE # 1:CC	LINE # 51:PP
LINE # 2:II	LINE # 52:PI
LINE # 5:OK	LINE # 53:IM
LINE # 10:CD	LINE # 60:KF
LINE # 12:OI	LINE # 70:NJ
LINE # 13:JJ	LINE # 71:PP
LINE # 14:AP	LINE # 100:MF
LINE # 15:IL	LINE # 102:FG
LINE # 16:BI	LINE # 103:NF
LINE # 17:JC	LINE # 104:NC
LINE # 30:CH	LINE # 105:HO
LINE # 32:IF	LINE # 106:OK
LINE # 33:IL	LINE # 107:KP
LINE # 34:FI	LINE # 108:FO
LINE # 35:NJ	LINE # 109:BB
LINE # 36:NB	LINE # 110:IK
LINE # 37:AJ	LINE # 120:BL
LINE # 38:IB	LINE # 122:CL
LINE # 39:AC	LINE # 400:AP
LINE # 40:HD	LINE # 402:OP
LINE # 41:NL	LINE # 403:AF
LINE # 42:NF	LINE # 404:PD
LINE # 43:BF	LINE # 405:OG
LINE # 44:EC	LINE # 420:CE
LINE # 46:CP	LINE # 422:AH
LINE # 47:FG	LINE # 424:KD
LINE # 48:FG	LINE # 500:MO
LINE # 50:EN	LINE # 501:FL

```
LINE # 502:BF
LINE # 504:FD
LINE # 505:CK
LINE # 506:DM
LINE # 507:CB
LINE # 510:CI
LINE # 1000:DF
LINE # 1002:AN
LINE # 1004:PF
LINE # 1005:PF
LINE # 1006:GG
LINE # 1007:LG
LINE # 1008:OH
LINE # 1009:CO
LINE # 1010:DG
LINE # 1020:EC
LINE # 1022:CF
LINE # 1024:CJ
LINE # 1025:IM
LINE # 1500:NA
LINE # 1502:HP
LINE # 1503:BB
LINE # 1504:JJ
LINE # 2003:EB
LINE # 2004:FO
LINE # 2006:IM
LINE # 2100:FI
LINE # 2102:KF
LINE # 2103:HL
LINE # 2104:AO
LINE # 2105:LL
LINE # 2106:DO
LINE # 2107:NI
LINE # 2108:CB
LINE # 10000:CM
LINE # 10001:BH
LINE # 10002:PJ
LINE # 10003:JF
LINE # 10004:NH
LINE # 10005:IH
LINE # 10006:OF
LINE # 10007:NO
LINE # 10008:GH
LINE # 10009:EM
LINE # 10010:GH
LINE # 10011:GH
LINE # 10013:PN
LINE # 10019:PN
LINE # 10020:OC
LINES: 105
```

BRISK

FROM PAGE 48

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```
• 2 PRINT"{SC}":GOSUB50000
• 3 PRINT"{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CR}{CR}{CR}{CR}WO
ULD YOU LIKE INSTRUCTIONS Y/N"
• 4 GET Q$:IFQ$="Y"THEN60000
• 5 IFQ$=""THEN4
• 6 PRINT"{SC}":POKE53281,1:POKE532
80,2
• 7 DIMP(2,39):DIMC$(39)
• 10 FORW=55296TO56295:POKEW,0:NEXT
W
• 11 SD=54272:FORL=SDTOSD+24:POKEL,
0:NEXT:POKESD+5,9:POKESD+6,0:POKE
SD+24,15:L=0
• 15 TM=0:TC=0:POKE1708+54272,4:POK
E1788+54272,4
• 17 FOR LO=1TO0STEP-1
•20 A=0:D=0:M=0:G=0:TY=0:TP=0:XY=0
:J=56039:AB=55687:AE=55689
•21 B=1509:N=1389
•37 GOSUB 1000:GOSUB 5000
•50 FOR Y=0TO39:C$(Y)="":NEXTY
•100 FORQ=0TO6
•101 A=A+1
•103 R=INT(RND(1)*40)
•105 IFC$(R)=STR$(R)THEN103
•110 C$(R)=STR$(R)
•115 IFQ=6THENFORK=0TO2:TR(K,1)=P(
K,R):NEXTK:GOSUB6000:T(9)=TR(2,1)
:GOTO130
•122 IFA=LOTHEND=D+1:FORK=0TO2:PL(
K,D)=P(K,R):NEXTK:GOSUB3000:GOTO1
30
•125 M=M+1:FORK=0TO2:MC(K,M)=P(K,R
):NEXTK:A=(LO-1)
•127 ON M GOSUB 1400,1500,1600
•130 NEXTQ:GOSUB 7000:GOSUB 9000
•131 WN=LO:IFWN=0THENGOSUB10000:GO
SUB8000:ZX=0:GOTO 133
•132 GOSUB8000:GOSUB10000
•133 IFWN=0THEN28000
•134 VL(1)=T(4):VL(2)=T0
•135 SU(1)=MC(2,I1):SU(2)=T(5)
•136 GS(1)=SU(1):IFSU(1)=T(9)THENS
U(1)=1:GOTO138
•137 SU(1)=0:
•138 GS(2)=SU(2):IFSU(2)=T(9)THENS
U(2)=1:GOTO140
•139 SU(2)=0
•140 GOSUB 12000
•141 GOSUB 18000
•142 G=0:IFAA=1THEN30000
•143 IFXY=1THEN250
•150 FOR X=7TO39
•151 PRINT"{HM}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{
CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}";X+1
152 G=G+1
•200 R=INT(RND(1)*40)
•210 IFC$(R)=STR$(R)THEN200
•220 C$(R)=STR$(R)
•230 IFTES=0THEN240
•233 FORK=0TO2:PL(K,I)=P(K,R):NEXT
K
•234 TES=0
•235 GOSUB14000
•236 IFX=39THEN27000
```

```
•237 GOTO 249
•240 FORK=0TO2:MC(K,I1)=P(K,R):NEX
 TK
•241 TES=1
•245 GOSUB 16000
•247 IFX=39THEN27000
•249 IFG=2THENXY=1:GOTO25000
•250 NEXTX
•300 POKE1340,32
•305 POKE1339,32
•306 POKE1415,32:POKE1497,32
•310 TM=TP:TC=TY
•315 POKE 1709,32:POKE1789,32
•330 PRINT"{HM}{CD}{CD}{CR}{CR}{CR
 }{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}
 THE SECOND HALF":RESTORE:TY=0:TP=
 0:AA=0:GOSUB7500:NEXTLO
 340 PRINT"{HM}{CD}{CD}{CD}{CD}{CD
 }{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
 {CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
 CD}{CD}{CR}{CR}IT'S A DRAW":GOTO4
 0000
•500 END
•700 POKE1472,66:POKE1512,66:POKE1
 552,66:POKE1468,66:POKE1508,66:PO
 KE1548,66
•720 POKE1432,73:POKE1431,67:POKE1
 430,67:POKE1429,67:POKE1428,85
•722 POKE1588,74:POKE1589,67:POKE1
 590,67:POKE1591,67:POKE1592,75:RE
 TURN
•750 POKE1514,66:POKE1518,66:POKE1
 474,66:POKE1554,66:POKE1434,85:PO
 KE1594,74
•752 POKE1435,67:POKE1436,67:POKE1
 437,67:POKE1438,73:POKE1478,66:PO
 KE1558,66
•754 POKE1595,67:POKE1596,67:POKE1
 597,67:POKE1598,75:RETURN
•760 POKE1520,66:POKE1524,66:POKE1
 480,66:POKE1560,66:POKE1484,66:PO
 KE1564,66
•762 POKE1444,73:POKE1443,67:POKE1
 442,67:POKE1441,67:POKE1440,85
•764 POKE1600,74:POKE1601,67:POKE1
 602,67:POKE1603,67:POKE1604,75:RE
 TURN
•800 POKE1272,66:POKE1312,66:POKE1
 352,66:POKE1268,66:POKE1308,66:PO
 KE1348,66
•820 POKE1232,73:POKE1231,67:POKE1
 230,67:POKE1229,67:POKE1228,85
•822 POKE1388,74:POKE1389,67:POKE1
 390,67:POKE1391,67:POKE1392,75:RE
 TURN
•850 POKE1314,66:POKE1318,66:POKE1
 274,66:POKE1354,66:POKE1234,85:PO
 KE1394,74
•852 POKE1235,67:POKE1236,67:POKE1
 237,67:POKE1238,73:POKE1278,66:PO
 KE1358,66
•854 POKE1395,67:POKE1396,67:POKE1
 397,67:POKE1398,75:RETURN
•860 POKE1320,66:POKE1324,66:POKE1
 280,66:POKE1360,66:POKE1284,66:PO
 KE1364,66
•862 POKE1244,73:POKE1243,67:POKE1
 242,67:POKE1241,67:POKE1240,85
•864 POKE1400,74:POKE1401,67:POKE1
 402,67:POKE1403,67:POKE1404,75:RE
 TURN
•960 POKE1800,66:POKE1804,66:POKE1
 760,66:POKE1840,66:POKE1764,66:PO
 KE1844,66
•962 POKE1724,73:POKE1723,67:POKE1
 722,67:POKE1721,67:POKE1720,85
•964 POKE1880,74:POKE1881,67:POKE1
 882,67:POKE1883,67:POKE1884,75:RE
 TURN
•970 POKE1806,66:POKE1810,66:POKE1
 766,66:POKE1846,66:POKE1770,66:PO
 KE1850,66
•972 POKE1730,73:POKE1729,67:POKE1
 728,67:POKE1727,67:POKE1726,85
•974 POKE1886,74:POKE1887,67:POKE1
 888,67:POKE1889,67:POKE1890,75:RE
 TURN
•1000 FORT=0TO39
•1005 FORU=0TO2
•1010 READ P(U,T)
•1015 NEXTU
•1020 NEXTT
•1030 RETURN
•1100 FORL=-1TO3:POKE1229+L,32:NEX
 T
•1105 FORL=-1TO3:POKE1269+L,32:NEX
 T
•1110 FORL=-1TO3:POKE1309+L,32:NEX
 T
•1120 FORL=-1TO3:POKE1349+L,32:NEX
 T
•1125 FORL=-1TO3:POKE1389+L,32:NEX
 T
•1130 RETURN
•1200 FORL=-1TO3:POKE1235+L,32:NEX
 T
```

```
1205 FORL=-1TO3:POKE1275+L,32:NEX
T
1210 FORL=-1TO3:POKE1315+L,32:NEX
T
1220 FORL=-1TO3:POKE1355+L,32:NEX
T
1225 FORL=-1TO3:POKE1395+L,32:NEX
T
1230 RETURN
1300 FORL=-1TO3:POKE1241+L,32:NEX
T
1305 FORL=-1TO3:POKE1281+L,32:NEX
T
1310 FORL=-1TO3:POKE1321+L,32:NEX
T
1320 FORL=-1TO3:POKE1361+L,32:NEX
T
1325 FORL=-1TO3:POKE1401+L,32:NEX
T
1330 RETURN
1400 GOSUB 800
1402 FORL=0TO2:POKE1269+L+54272,6
:POKE1269+L,102:NEXT
1410 FORL=0TO2:POKE1309+L+54272,6
:POKE1309+L,102:NEXT
1420 FORL=0TO2:POKE1349+L+54272,6
:POKE1349+L,102:NEXT
1430 RETURN
1500 GOSUB 850
1502 FORL=0TO2:POKE1275+L+54272,6
:POKE1275+L,102:NEXT
1510 FORL=0TO2:POKE1315+L+54272,6
:POKE1315+L,102:NEXT
1520 FORL=0TO2:POKE1355+L+54272,6
:POKE1355+L,102:NEXT
1530 RETURN
1600 GOSUB 860
1602 FORL=0TO2:POKE1281+L+54272,6
:POKE1281+L,102:NEXT
1610 FORL=0TO2:POKE1321+L+54272,6
:POKE1321+L,102:NEXT
1620 FORL=0TO2:POKE1361+L+54272,6
:POKE1361+L,102:NEXT
1630 RETURN
2000 DATA 32,1,65,32,50,65,32,51,
65,32,52,65,32,53,65,32,54,65,32,
55,65
2010 DATA 32,10,65,32,17,65,32,11
,65
2020 DATA 32,1,83,32,50,83,32,51,
83,32,52,83,32,53,83,32,54,83,32,
55,83
2030 DATA 32,10,83,32,17,83,32,11
,83
2040 DATA 32,1,88,32,50,88,32,51,
88,32,52,88,32,53,88,32,54,88,32,
55,88
2050 DATA 32,10,88,32,17,88,32,11
,88
2060 DATA 32,1,90,32,50,90,32,51,
90,32,52,90,32,53,90,32,54,90,32,
55,90
2070 DATA 32,10,90,32,17,90,32,11
,90
3000 ON D GOSUB 700,750,760
3001 IFPL(2,D)=90ORPL(2,D)=83THEN
GOSUB11000:GOTO3005
3002 POKE(B-40+54272),0:POKEB+1+5
4272,0:POKEB+42+54272,0
3005 POKEB-40,PL(2,D):FORL=1TO1:P
OKEB+L,PL(L,D):NEXTL:POKEB+42,PL(
2,D):B=B+6
3010 RETURN
5000 POKE 1630,49:POKE1636,50:POK
E1642,51
5010 POKE 1194,20:POKE1195,8:POKE
1196,5:POKE 1198,4:POKE1199,5:POK
E1200,1
5020 POKE1201,12
5030 POKE1710,25:POKE1711,15:POKE
1712,21:POKE1713,18
5040 POKE1715,16:POKE1716,12:POKE
1717,1:POKE1718,25:POKE1708,81
5050 POKE1449,20:POKE1450,18:POKE
1451,21:POKE1452,13:POKE1453,16
5100 POKE1529,19:POKE1530,3:POKE1
531,15:POKE1532,18:POKE1533,5
5110 POKE1608,25:POKE1609,15:POKE
1610,21
5120 POKE1613,13:POKE1614,5:QB=13
26:QV=54272
5130 POKEQB+QV,0:POKEQB,3:POKEQB+
1+QV,0:POKEQB+1,1:POKEQB+2+QV,0:P
OKEQB+2,18
5131 POKEQB+3+QV,0:POKEQB+3,4:POK
EQB+4+QV,0:POKEQB+4,19
5132 POKEQB+6+QV,0:POKEQB+6,16:PO
KEQB+7+QV,0:POKEQB+7,12:POKEQB+8+
QV,0
5133 POKEQB+8,1:POKEQB+9+QV,0:POK
EQB+9,25:POKEQB+10+QV,0:POKEQB+10
,5
5134 POKEQB+11,0:POKEQB+11,4
5200 RETURN
6000 GOSUB6100
6001 IFTR(2,1)=90ORTR(2,1)=83THEN
```

```
POKE55687,10:POKE55728,10:POKE557
69,10
6005 POKE1415,TR(2,1)
6008 FORL=1TO1:POKE1455+L,TR(L,1)
:NEXTL:POKE1455+42,TR(2,1)
6010 RETURN
6100 POKE1498,66:POKE1458,66:POKE
1418,66:POKE1378,73:POKE1538,75
6105 POKE1377,67:POKE1376,67:POKE
1375,67:POKE1537,67:POKE1536,67:P
OKE1535,67
6110 POKE1534,74:POKE1494,66:POKE
1454,66:POKE1414,66:POKE1374,85
6190 RETURN
7000 POKE 1194,32:POKE1195,32:POK
E1196,32:POKE1198,32:POKE1199,32:
POKE1200,32
7001 POKE 1201,32
7010 POKE 1193,12:POKE1194,5:POKE
1195,20:POKE1196,39:POKE1197,19
7020 POKE1199,16:POKE1200,12:POKE
1201,1:POKE1202,25:RETURN
7500 POKE1193,32:POKE1194,32:POKE
1195,32:POKE1196,32:POKE1197,32:P
OKE1199,32
7501 POKE55687,0:POKE55728,0:POKE
55769,0
7502 POKE1200,32:POKE1201,32:POKE
1202,32
7503 XA=0:XB=0:XC=0:X=0
7504 POKE1194,19:POKE1195,8:POKE1
196,21:POKE1197,6:POKE1198,6:POKE
1199,12
7506 POKE1200,5:FORRY=1TO4000:NEX
TRY
7510 POKE1193,32:POKE1194,32:POKE
1195,32:POKE1196,32:POKE1197,32:P
OKE1198,32
7512 POKE1199,32:POKE1200,32:GOSU
B14100:GOSUB14200:GOSUB14300:RETU
RN
7800 IFT(1)>125ANDT(6)<>T(9)THEN1
0270
7810 IFT(2)>125ANDT(7)<>T(9)THEN1
0271
7820 IFT(3)>125ANDT(8)<>T(9)THEN1
0272
7830 GOTO 10261
8000 GET I$:IFI$=""THEN8000
8001 IFVAL(I$)<1ORVAL(I$)>3THEN80
00
8002 I=VAL(I$):IF(I=XA)OR(I=XB)OR
(I=XC)THEN8000
8003 POKESD+1,(15*I):POKESD,120:P
OKESD+4,17:FORT=1TO100:NEXT:POKES
D+4,16
8004 IFPL(2,I)=90ORPL(2,I)=83THEN
FORL=JTOJ+82STEP41:POKEL-6,10:NEX
T:GOTO8006
8005 FORL=JTOJ+82STEP41:POKEL-6,0
:NEXT
8006 REM
8007 IFI=1THENGOSUB8100:GOTO8012
8008 IFI=2THENGOSUB8200:GOTO8012
8009 IFI=3THENGOSUB8300
8012 T0=0
8013 GOSUB960:POKE1761,PL(2,I):FO
RK=1TO1:POKE1802,PL(K,I):NEXT:POK
E1843,PL(2,I)
8015 FORK=0TO1:T0=T0+PL(K,I):NEXT
K
8016 T(5)=PL(2,I)
8017 IFT0=33THENT0=150:F=11:GOTO8
040
8018 IFT0=83THENT0=125:F=10:GOTO8
040
8019 IFT0=43THENT0=115:F=4:GOTO80
40
8020 IFT0=42THENT0=110:F=3:GOTO80
40
8021 IFT0=49THENT0=100:F=2:GOTO80
40
8030 F=0
8040 GOTO8900
8100 FORL=-1TO3:POKE1429+L,32 :NE
XT
8105 FORL=-1TO3:POKE1469+L,32 :NE
XT
8110 FORL=-1TO3:POKE1509+L,32 :NE
XT
8120 FORL=-1TO3:POKE1549+L,32 :NE
XT
8125 FORL=-1TO3:POKE1589+L,32 :NE
XT
8130 RETURN
8200 FORL=-1TO3:POKE1435+L,32 :NE
XT
8205 FORL=-1TO3:POKE1475+L,32 :NE
XT
8210 FORL=-1TO3:POKE1515+L,32 :NE
XT
8220 FORL=-1TO3:POKE1555+L,32 :NE
XT
8225 FORL=-1TO3:POKE1595+L,32 :NE
XT
8230 RETURN
```

```
8300 FORL=-1TO3:POKE1441+L,32 :NE
XT
8305 FORL=-1TO3:POKE1481+L,32 :NE
XT
8310 FORL=-1TO3:POKE1521+L,32 :NE
XT
8320 FORL=-1TO3:POKE1561+L,32 :NE
XT
8325 FORL=-1TO3:POKE1601+L,32 :NE
XT
8330 RETURN
8900 IFX=>39THENGOSUB62500
8999 RETURN
9000 FL=0
9010 IFPL(2,1)=TR(2,1)THENFL=1:GO
TO9999
9020 IFPL(2,2)=TR(2,1)THENFL=1:GO
TO9999
9030 IFPL(2,3)=TR(2,1)THENFL=1
9999 RETURN
10000 FORV=1TO3
10100 FORK=0TO1
10150 T(V)=T(V)+MC(K,V)
10200 NEXTK
10205 T(5+V)=MC(2,V)
10210 IFT(V)=33THENT(V)=150:GOTO1
0255
10220 IFT(V)=83THENT(V)=125:GOTO1
0255
10230 IFT(V)=43THENT(V)=115:GOTO1
0255
10240 IFT(V)=42THENT(V)=110:GOTO1
0255
10250 IFT(V)=49THENT(V)=100
10255 NEXTV:GOSUB9000
10256 IFZX=0ANDFL=0ANDX<39THEN780
0
10261 IFZX=0THEN10301
10262 IF(T(5)=T(9))THEN10300
10263 SE(1)=0:SE(2)=0:SE(3)=0
10264 IFT(1)>T0ANDT(6)=T(5)THENSE
(1)=T(1)
10265 IFT(2)>T0ANDT(7)=T(5)THENSE
(2)=T(2)
10267 IFT(3)>T0ANDT(8)=T(5)THENSE
(3)=T(3)
10268 IFSE(1)>0ORSE(2)>0ORSE(3)>0
THEN10970
10269 GOTO10277
10270 I1=1:T(4)=T(1):GOTO10705
10271 I1=2:T(4)=T(2):GOTO10705
10272 I1=3:T(4)=T(3):GOTO10705
10277 IFT0>=125THEN10280
```

```
10278 IF(T0=115)AND(X>20ANDX<36)T
HEN10280
10279 GOTO 10300
10280 SY(1)=200:SY(2)=200:SY(3)=2
00
10281 IFT(6)=T(9)THENSY(1)=T(1)
10282 IFT(7)=T(9)THENSY(2)=T(2)
10284 IFT(8)=T(9)THENSY(3)=T(3)
10285 IFSY(1)<2000RSY(2)<2000RSY(
3)<200THEN10990
10286 GOTO 10301
10300 IFT0=125ANDT(5)=T(9)THEN102
63
10301 IFT(1)<=T(2)THEN10400
10310 IFT(2)<=T(3)THEN10500
10320 T(4)=T(3):I1=3:GOTO 10700
10400 IFT(1)<=T(3)THENT(4)=T(1):I
1=1:GOTO10700
10410 T(4)=T(3):I1=3:GOTO 10700
10500 T(4)=T(2):I1=2:GOTO10700
10700 IFT(5+I1)=T(9)THEN10930
10705 IFT(4)=150THENO=11:GOTO1090
0
10720 IFT(4)=125THENO=10:GOTO1090
0
10740 IFT(4)=115THENO=4:GOTO10900
10760 IFT(4)=110THENO=3:GOTO10900
10780 IFT(4)=100THENO=2:GOTO10900
10800 O=0
10900 IFI1=1THENGOSUB1100:GOTO109
15
10905 IFI1=2THENGOSUB1200:GOTO109
15
10910 IFI1=3THENGOSUB1300
10915 POKE1790,13:POKE1791,25:POK
E1793,16:POKE1794,12:POKE1795,1:P
OKE1796,25
10916 POKESD+1,(15*I1):POKESD,120
:POKESD+4,17:FORT=1TO100:NEXT:POK
ESD+4.16
10918 IFMC(2,I1)=90ORMC(2,I1)=83T
HENFORL=JTOJ+82STEP41:POKEL,10:NE
XT:GOTO10920
10919 FORL=JTOJ+82STEP41:POKEL,0:
NEXT
10920 GOSUB970
10922 POKE1767,MC(2,I1):FORK=1TO1
:POKE1808,MC(K,I1):NEXT:POKE1849,
MC(2,I1)
10925 GOTO10999
10930 XX=I1:IFX=39THEN10705
10931 BB=6-I1
10932 IFBB=5THEN10940
```

```
10933 IFBB=4THEN10950
10934 IFBB=3THEN10960
10940 IFT(2)>T(3)THENI1=3:T(4)=T(
3):GOTO10944
10942 I1=2:T(4)=T(2)
10944 IFT(5+I1)=T(9)ORT(4)>115THE
NI1=XX:T(4)=T(I1):GOTO10705
10946 GOTO10705
10950 IFT(1)>T(3)THENI1=3:T(4)=T(
3):GOTO10954
10952 I1=1:T(4)=T(1)
10954 IFT(5+I1)=T(9)ORT(4)>115THE
NI1=XX:T(4)=T(I1):GOTO10705
10956 GOTO10705
10960 IFT(1)>T(2)THENI1=2:T(4)=T(
2):GOTO10964
10962 I1=1:T(4)=T(1)
10964 IFT(5+I1)=T(9)ORT(4)>115THE
NI1=XX:T(4)=T(I1):GOTO10705
10966 GOTO10705
10970 IFSE(1)>SE(2)THEN10980
10974 IFSE(2)>SE(3)THEN10985
10976 GOTO 10272
10980 IFSE(1)>SE(3)THEN10270
10982 GOTO 10272
10985 GOTO 10271
10990 IFSY(1)<SY(2)THEN10993
10991 IFSY(2)<SY(3)THEN10995
10992 GOTO 10272
10993 IFSY(1)<SY(3)THEN10270
10994 GOTO 10272
10995 GOTO 10271
10999 RETURN
11000 POKE(B-40+54272),10:POKEB+1
+54272,10:POKEB+42+54272,10
11010 RETURN
12000 IFSU(1)>SU(2)THEN 12500
12100 IFSU(1)<SU(2)THEN 12600
12200 IFVL(1)>VL(2)ANDGS(1)=GS(2)
THEN12500
12300 GOTO12600
12400 IFAA=1THENFORK=0TO2:MC(K,I1
)=100:NEXT
12402 RETURN
12500 IFWN=0THENGOSUB21000:GOTO12
550
12525 GOSUB20000
12550 GOTO 12400
12600 IFWN=0THENGOSUB20000:GOTO12
650
12625 GOSUB21000
12650 GOTO 12400
14000 B=1509
14005 ON I GOSUB 700,750,760
14010 IFI=1THEN B=B+0:GOSUB14100:
GOTO14040
14020 IFI=2THEN B=B+6:GOSUB14200:
GOTO14040
14030 IFI=3THEN B=B+12:GOSUB14300
14040 IFPL(2,I)=90ORPL(2,I)=83THE
ND=I:GOSUB11000:GOTO14046
14042 POKEB-40+54272,0:POKEB+1+54
272,0:POKEB+42+54272,0
14046 POKEB-40,PL(2,I):FORL=1TO1:
POKEB+L,PL(L,I):NEXTL:POKEB+42,PL
(2,I)
14050 RETURN
14100 FORL=0TO2:POKE1469+L,32:NEX
T
14110 FORL=0TO2:POKE1509+L,32:NEX
T
14120 FORL=0TO2:POKE1549+L,32:NEX
T
14130 RETURN
14200 FORL=0TO2:POKE1475+L,32:NEX
T
14210 FORL=0TO2:POKE1515+L,32:NEX
T
14220 FORL=0TO2:POKE1555+L,32:NEX
T
14230 RETURN
14300 FORL=0TO2:POKE1481+L,32:NEX
T
14310 FORL=0TO2:POKE1521+L,32:NEX
T
14320 FORL=0TO2:POKE1561+L,32:NEX
T
14330 RETURN
16000 N=1389
16010 IFI1=1THEN N=N+0:GOSUB1400:
GOTO16040
16020 IFI1=2THEN N=N+6:GOSUB1500:
GOTO16040
16030 IFI1=3THEN N=N+12:GOSUB1600
16040 REM
16050 RETURN
18000 FORK=1TO8:T(K)=0:NEXTK
18100 FORK=1TO1000:NEXTK
18140 FORL=-1TO3:POKE1721+L,32:PO
KE1727+L,32:NEXT
18150 FORL=-1TO3:POKE1761+L,32:PO
KE1767+L,32:NEXT
18160 FORL=-1TO3:POKE1801+L,32:PO
KE1807+L,32:NEXT
18170 FORL=-1TO3:POKE1841+L,32:PO
KE1847+L,32:NEXT
```

```
18180 FORL=-1TO3:POKE1881+L,32:PO
KE1887+L,32:NEXT
•18201 RETURN
•20000 REM
•20050 TP=TP+O+F
•20060 POKE1788,81:POKE1708,32:TES
=0:ZX=0:WN=0
•20102 IFZZ=1THEN20110
•20103 IFTP+TM>120THENPRINT"{HM}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}{CR}{CR}{
CR}I WIN I'M THE BEST!":ZZ=1
•20104 PRINT"{HM}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{
CR}{CR}{CR}{CR}{CR}{CR}";TP+TM
•20105 IFZZ=1THEN40000
20110 RETURN
21000 REM
21050 TY=TY+O+F
21060 POKE1708,81:POKE1788,32:TES
=1:ZX=1:WN=1
•21102 IFZZ=1THEN21110
•21103 IFTY+TC>120THENPRINT"{HM}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CR}{CR}YOU W
IN YOU GOT LUCKY!":ZZ=1
•21104 PRINT"{HM}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{
CR}{CR}";TY+TC
•21105 IFZZ=1THEN40000
•21110 RETURN
•25000 IFZX=0THEN25500
•25010 GOTO 26000
•25500 GOSUB10000:GOSUB8000:GOTO13
3
•26000 GOSUB8000:GOSUB10000:GOTO13
3
•27000 FORL=0TO1:POKE1455+L,32:NEX
TL
•27005 IFTES=0THENFORK=0TO2:MC(K,I
1)=TR(K,1):NEXTK:GOSUB16000:AA=1:
GOTO30000
•27010 FORK=0TO2:PL(K,I)=TR(K,1):N
EXTK:GOSUB14000:AA=1:GOTO30000
•28000 VL(1)=T0:VL(2)=T(4)
•28010 SU(1)=T(5):SU(2)=MC(2,I1)
•28020 GS(1)=SU(1):IFSU(1)=T(9)THE
NSU(1)=1:GOTO28040
•28030 SU(1)=0
•28040 GS(2)=SU(2):IFSU(2)=T(9)THE
NSU(2)=1:GOTO140
•28050 SU(2)=0:GOTO140
•30000 IF(TY+TP=120)THEN300
•30001 IFZX=0THEN30500
•30100 GOTO 30600
•30500 GOSUB10000:GOSUB8000:GOTO13
3
•30600 GOSUB8000:GOSUB10000:GOTO13
3
•40000 PRINT"{HM}{CD}{CD}{CD}{CR}{
CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}{CR}{CR}{CR}GAME OVE
R"
•40002 PRINT"{HM}{CD}{CD}{CD}{CD}{
CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR
}{CR}{CR}{CR}{CR}{CR}{CR}{CR}TRY
AGAIN (Y/N)"
•40003 GETQ$:IFQ$="Y"THENRESTORE:P
RINT"{SC}":GOTO62000
•40004 FORZ=0TO15:POKE53280,Z:NEXT
•40005 GOTO40003
•50000 PRINT"{HM}{CD}{CD}{CD}{CR}{
CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}{CR}{CR}B R I S
 K"
•50010 PRINT"{HM}{CD}{CD}{CD}{CD}{
CD}{CD}{CR}{CR}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}DESIGNED BY JOSEPH
FUSCO "
•50060 PRINT"{HM}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CR}{CR
}{CR}COPYRIGHT 1984 BY JOSEPH FUS
CO"
•50070 PRINT"{HM}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{C
D}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CR
}{CR}{CR}ALL RIGHTS RESERVED."
•50998 FOR WA=1TO6000:NEXT:PRINT"{
SC}"
•50999 RETURN
•60000 PRINT"{SC}":PRINT"{HM}{CD}{
CR}OBJECT: TO BE THE FIRST TO SCO
RE 121 P-          OINTS."
•60002 PRINT"          THERE ARE TW
O HANDS PER-GAME."
```

```
60005 PRINT"{CR}{CR}TRUMP: THE GA
 ME BEGINS WITH THREE ";
60006 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}{CR}{CR}{CR}{CR}CA
 RDS DEALT TO EACH PLAYER   "
60007 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR} THE SEVENTH CARD IS D
 EALT THE  ";
60008 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}SUIT OF THIS CARD
 IS TRUMP"
60009 PRINT"{CR}{CR}{CR}{CR}EX/:
 TRUMP-CARD 2X CLUBS IS TRUMP!"
60010 PRINT"{CR}{CR}{CR}PLAY: IF
 YOU LEAD YOUR OPPONENT CAN "
60011 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}FOLLOW SUIT,LEASH
 OR TRUMP."
60012 PRINT"{CR}{CR}{CR}{CR}EX/:
 YOUR PLAY KZ,COMPUTER PLAYS 6Z ";
60013 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}YOU WIN BECAUSE TH
 E KZ > 6Z."
60014 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}IF YOU PLAY THE AS
  AND CLUBS IS";
60015 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR} TRUMP THE COMPUTER CA
 N PLAY THE";
60016 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR} 4X TO TRUMP YOUR ACE
 AND WIN."
60017 PRINT"{CR}{CR}{CR}NOTE: TRY
  TO LET THE COMPUTER LEAD AS";
60018 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR} OFTEN AS POSSIBLE SO
 YOU CAN";
60019 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}{CR}{CR}{CR}PLAY T
 HE HIGHER CARD IN THAT";
60020 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}{CR}{CR} SUIT OR T
 RUMP A LOAD TO WIN"
60021 PRINT"{CR}{CR}{CR}LOAD: ACE
 S OR THREES"
60022 PRINT"{CR}{CR}VALUE: A=11,3
 =10,K=4,J=3,Q=2 POINTS";
60023 PRINT"{CR}{CR}{CR}{CR}{CR}{
 CR}{CR}{CR}{CR}{CR}{CR}{CR}2,4,5,
 6,7 HAVE NO POINT VALUE"
60040 PRINT"{CR}{CR}PRESS Y TO PL
 AY,E FOR MORE EXAMPLES"
60041 GETQ$:IFQ$="Y"THEN6
60042 IFQ$="E"THEN60100
60043 GOTO60041
60100 PRINT"{SC}{HM}{CD}{CR}EX/:F
 OLLOW SUIT"
60104 PRINT"{CR}COMPUTER PLAYS KZ
  YOU PLAY QZ"
60107 PRINT"{CR}COMPUTER WINS."
60109 PRINT
60111 PRINT"{CR}EX/:FOLLOW SUIT H
 IGHER CARD"
60113 PRINT"{CR}COMPUTER PLAYS 7X
  YOU PLAY AX"
60115 PRINT"{CR}YOU WIN YOU PLAYE
 D A GREATER VALUE CARD";
60116 PRINT"{CR}AND THE SAME SUIT
 "
60117 PRINT
60119 PRINT"{CR}EX/:LEASH PLAY,TO
  PLAY YOUR LOWEST CARD";
60121 PRINT"{CR}COMPUTER PLAYS JA
  YOU PLAY 4Z"
60123 PRINT"{CR}COMPUTER WINS YOU
  DID NOT FOLLOW SUIT"
60125 PRINT"{CR}YOU DID NOT TRUMP
 "
60128 PRINT
60130 PRINT"{CR}EX/:TRUMP"
60132 PRINT"{CR}COMPUTER PLAYS AX
  TRUMP IS S YOU PLAY"
60134 PRINT"{CR}THE 2S YOU WIN YO
 U TRUMPED THE ACE"
60150 PRINT
60152 PRINT"{CR}PRESS P TO PLAY"
60155 GETQ$:IFQ$="P"THEN6
60156 GOTO 60155
62000 RUN6
62500 IFI=1THENXA=1:GOTO62550
62510 IFI=2THENXB=2:GOTO62550
62520 IFI=3THENXC=3
62550 RETURN
```

BUG REPELLENT LINE CODES FOR BRISK

LINE # 2:CN	LINE # 20:NG
LINE # 3:MG	LINE # 21:OE
LINE # 4:GM	LINE # 37:OF
LINE # 5:NE	LINE # 50:FB
LINE # 6:IA	LINE # 100:JM
LINE # 7:PC	LINE # 101:BF
LINE # 10:PN	LINE # 103:EC
LINE # 11:GF	LINE # 105:CM
LINE # 15:LA	LINE # 110:DB
LINE # 17:GE	LINE # 115:OH

```
LINE # 122:JE
LINE # 125:LN
LINE # 127:EN
LINE # 130:AP
LINE # 131:LG
LINE # 132:AO
LINE # 133:NG
LINE # 134:HF
LINE # 135:MO
LINE # 136:JB
LINE # 137:FB
LINE # 138:BI
LINE # 139:BK
LINE # 140:IG
LINE # 141:JA
LINE # 142:PA
LINE # 143:GF
LINE # 150:OJ
LINE # 151:KL
LINE # 152:CF
LINE # 200:EC
LINE # 210:DE
LINE # 220:DB
LINE # 230:KG
LINE # 233:JL
LINE # 234:NE
LINE # 235:IM
LINE # 236:KO
LINE # 237:CN
LINE # 240:OD
LINE # 241:NL
LINE # 245:IK
LINE # 247:KO
LINE # 249:OA
LINE # 250:NK
LINE # 300:AL
LINE # 305:BD
LINE # 306:EC
LINE # 310:CK
LINE # 315:DG
LINE # 330:BP
LINE # 340:NN
LINE # 500:IC
LINE # 700:IJ
LINE # 720:AO
LINE # 722:MI
LINE # 750:NE
LINE # 752:HG
LINE # 754:ED
LINE # 760:CG
LINE # 762:AD
LINE # 764:CG
LINE # 800:AN
LINE # 820:KM
LINE # 822:JG
LINE # 850:DI
LINE # 852:NC
LINE # 854:OP
LINE # 860:CC
LINE # 862:OJ
LINE # 864:AM
LINE # 960:CO
LINE # 962:NA
LINE # 964:DI
LINE # 970:EN
LINE # 972:GH
LINE # 974:JJ
LINE # 1000:NG
LINE # 1005:IM
LINE # 1010:BO
LINE # 1015:NJ
LINE # 1020:NG
LINE # 1030:IM
LINE # 1100:BJ
LINE # 1105:DN
LINE # 1110:DE
LINE # 1120:BI
LINE # 1125:DM
LINE # 1130:IM
LINE # 1200:BM
LINE # 1205:GA
LINE # 1210:PD
LINE # 1220:BH
LINE # 1225:DL
LINE # 1230:IM
LINE # 1300:BL
LINE # 1305:DP
LINE # 1310:OO
LINE # 1320:BC
LINE # 1325:CF
LINE # 1330:IM
LINE # 1400:CI
LINE # 1402:LF
LINE # 1410:AH
LINE # 1420:OP
LINE # 1430:IM
LINE # 1500:DF
LINE # 1502:OL
LINE # 1510:PB
LINE # 1520:OJ
LINE # 1530:IM
LINE # 1600:DC
LINE # 1602:GJ
LINE # 1610:EL
LINE # 1620:JD
LINE # 1630:IM
LINE # 2000:AH
LINE # 2010:JH
LINE # 2020:LL
LINE # 2030:IL
LINE # 2040:GI
LINE # 2050:CK
LINE # 2060:FJ
LINE # 2070:HP
LINE # 3000:KM
LINE # 3001:OH
LINE # 3002:EA
LINE # 3005:JH
LINE # 3010:IM
LINE # 5000:IK
LINE # 5010:BC
LINE # 5020:AD
LINE # 5030:EC
LINE # 5040:DN
LINE # 5050:JF
LINE # 5100:BJ
LINE # 5110:BG
LINE # 5120:OO
LINE # 5130:PK
LINE # 5131:NO
LINE # 5132:MH
LINE # 5133:GM
LINE # 5134:GI
LINE # 5200:IM
LINE # 6000:FA
LINE # 6001:IL
LINE # 6005:AI
LINE # 6008:CD
LINE # 6010:IM
LINE # 6100:LO
LINE # 6105:KM
LINE # 6110:LI
LINE # 6190:IM
LINE # 7000:GK
LINE # 7001:AB
LINE # 7010:HM
LINE # 7020:FL
LINE # 7500:AG
LINE # 7501:JN
LINE # 7502:GI
LINE # 7503:HC
LINE # 7504:FF
LINE # 7506:JL
LINE # 7510:BB
LINE # 7512:CG
LINE # 7800:IN
LINE # 7810:NO
LINE # 7820:CL
LINE # 7830:IP
LINE # 8000:MO
LINE # 8001:MO
LINE # 8002:IA
LINE # 8003:MG
LINE # 8004:EO
LINE # 8005:PH
LINE # 8006:JD
LINE # 8007:FJ
LINE # 8008:ID
LINE # 8009:MA
LINE # 8012:GL
LINE # 8013:CN
LINE # 8015:LA
LINE # 8016:CF
LINE # 8017:OI
LINE # 8018:AM
LINE # 8019:JA
LINE # 8020:KP
LINE # 8021:HO
LINE # 8030:CL
LINE # 8040:GC
LINE # 8100:DH
LINE # 8105:DL
LINE # 8110:CK
LINE # 8120:AO
LINE # 8125:DC
LINE # 8130:IM
LINE # 8200:DC
LINE # 8205:DG
LINE # 8210:GB
LINE # 8220:CF
LINE # 8225:GJ
LINE # 8230:IM
LINE # 8300:GJ
LINE # 8305:GN
LINE # 8310:EE
LINE # 8320:GI
LINE # 8325:BL
LINE # 8330:IM
LINE # 8900:OB
LINE # 8999:IM
LINE # 9000:HJ
LINE # 9010:IP
LINE # 9020:OG
LINE # 9030:GN
LINE # 9999:IM
LINE # 10000:JL
LINE # 10100:ID
LINE # 10150:BK
LINE # 10200:MP
LINE # 10205:BL
LINE # 10210:FN
LINE # 10220:HI
```

```
LINE # 10230:CN
LINE # 10240:PH
LINE # 10250:MK
LINE # 10255:HF
LINE # 10256:LL
LINE # 10261:MI
LINE # 10262:OB
LINE # 10263:FA
LINE # 10264:IN
LINE # 10265:BL
LINE # 10267:DB
LINE # 10268:GN
LINE # 10269:JG
LINE # 10270:CM
LINE # 10271:IK
LINE # 10272:PI
LINE # 10277:KA
LINE # 10278:LG
LINE # 10279:ID
LINE # 10280:MN
LINE # 10281:IN
LINE # 10282:BE
LINE # 10284:GD
LINE # 10285:EC
LINE # 10286:IC
LINE # 10300:DO
LINE # 10301:DF
LINE # 10310:CK
LINE # 10320:CJ
LINE # 10400:JC
LINE # 10410:CJ
LINE # 10500:CJ
LINE # 10700:MC
LINE # 10705:GB
LINE # 10720:IK
LINE # 10740:IB
LINE # 10760:GL
LINE # 10780:FB
LINE # 10800:DA
LINE # 10900:LF
LINE # 10905:KH
LINE # 10910:NN
LINE # 10915:HN
LINE # 10916:ED
LINE # 10918:GJ
LINE # 10919:DB
LINE # 10920:DA
LINE # 10922:OD
LINE # 10925:KB
LINE # 10930:ML
LINE # 10931:JE
LINE # 10932:KC
LINE # 10933:JA
LINE # 10934:JC
LINE # 10940:MC
LINE # 10942:OO
LINE # 10944:EN
LINE # 10946:JK
LINE # 10950:CA
LINE # 10952:BE
LINE # 10954:EN
LINE # 10956:JK
LINE # 10960:PG
LINE # 10962:BE
LINE # 10964:EN
LINE # 10966:JK
LINE # 10970:FA
LINE # 10974:CL
LINE # 10976:JD
LINE # 10980:GL
LINE # 10982:JD
LINE # 10985:JA
LINE # 10990:CM
LINE # 10991:FC
LINE # 10992:JD
LINE # 10993:AB
LINE # 10994:JD
LINE # 10995:JA
LINE # 10999:IM
LINE # 11000:FL
LINE # 11010:IM
LINE # 12000:IC
LINE # 12100:FF
LINE # 12200:NK
LINE # 12300:II
LINE # 12400:KN
LINE # 12402:IM
LINE # 12500:DA
LINE # 12525:IP
LINE # 12550:IG
LINE # 12600:AE
LINE # 12625:IC
LINE # 12650:IG
LINE # 14000:ML
LINE # 14005:ID
LINE # 14010:LA
LINE # 14020:GK
LINE # 14030:FM
LINE # 14040:MP
LINE # 14042:BE
LINE # 14046:CB
LINE # 14050:IM
LINE # 14100:KA
LINE # 14110:FN
LINE # 14120:KJ
LINE # 14130:IM
LINE # 14200:IL
LINE # 14210:GI
LINE # 14220:IE
LINE # 14230:IM
LINE # 14300:JG
LINE # 14310:GD
LINE # 14320:KP
LINE # 14330:IM
LINE # 16000:NN
LINE # 16010:CC
LINE # 16020:BA
LINE # 16030:IK
LINE # 16040:JD
LINE # 16050:IM
LINE # 18000:KF
LINE # 18100:AO
LINE # 18140:FJ
LINE # 18150:LJ
LINE # 18160:PH
LINE # 18170:KH
LINE # 18180:IH
LINE # 18201:IM
LINE # 20000:JD
LINE # 20050:PB
LINE # 20060:JH
LINE # 20102:MI
LINE # 20103:CM
LINE # 20104:LB
LINE # 20105:MO
LINE # 20110:IM
LINE # 21000:JD
LINE # 21050:ON
LINE # 21060:JK
LINE # 21102:MJ
LINE # 21103:OI
LINE # 21104:KI
LINE # 21105:MO
LINE # 21110:IM
LINE # 25000:NP
LINE # 25010:IF
LINE # 25500:IJ
LINE # 26000:EB
LINE # 27000:MO
LINE # 27005:KM
LINE # 27010:OH
LINE # 28000:BH
LINE # 28010:EC
LINE # 28020:PG
LINE # 28030:BN
LINE # 28040:BI
LINE # 28050:FM
LINE # 30000:JB
LINE # 30001:MJ
LINE # 30100:IE
LINE # 30500:IJ
LINE # 30600:EB
LINE # 40000:IF
LINE # 40002:HG
LINE # 40003:KL
LINE # 40004:NI
LINE # 40005:IG
LINE # 50000:CD
LINE # 50010:OM
LINE # 50060:NN
LINE # 50070:NO
LINE # 50998:OA
LINE # 50999:IM
LINE # 60000:LJ
LINE # 60002:CC
LINE # 60005:KA
LINE # 60006:MO
LINE # 60007:AE
LINE # 60008:HI
LINE # 60009:FC
LINE # 60010:BO
LINE # 60011:GJ
LINE # 60012:JF
LINE # 60013:KB
LINE # 60014:LD
LINE # 60015:DC
LINE # 60016:OA
LINE # 60017:GD
LINE # 60018:NE
LINE # 60019:LA
LINE # 60020:GI
LINE # 60021:JA
LINE # 60022:JP
LINE # 60023:LC
LINE # 60040:GJ
LINE # 60041:JA
LINE # 60042:OH
LINE # 60043:IK
LINE # 60100:PF
LINE # 60104:IJ
LINE # 60107:OI
LINE # 60109:JJ
LINE # 60111:KC
LINE # 60113:KJ
LINE # 60115:HP
LINE # 60116:AH
LINE # 60117:JJ
LINE # 60119:DB
LINE # 60121:AM
LINE # 60123:AA
LINE # 60125:GN
LINE # 60128:JJ
```

```
LINE # 60130:KF    LINE # 62000:MA
LINE # 60132:JL    LINE # 62500:II
LINE # 60134:HB    LINE # 62510:MD
LINE # 60150:JJ    LINE # 62520:NK
LINE # 60152:BF    LINE # 62550:IM
LINE # 60155:HL    LINES: 456
LINE # 60156:II
```

RANDOM DISPLAY

FROM PAGE 95

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```
10 GOSUB 500:GOSUB 600
100 GOSUB 640
110 IF PEEK(197)=64 THEN 110
120 GOTO 100
497 REM
498 REM SET UP CHARACTER SET--64
VERSION
499 REM
500 CH=1024*14:FOR I=CH TO CH+135
:READ N:POKE I,N:NEXT
501 REM VIC VERSION: 500 CH=7168:
FOR I=CH TO CH+135:READ N:POKE I,
N:NEXT
502 REM IF YOU DON'T USE CUSTOM C
HARACTERS, REPLACE 500-511 WITH:
500 REM
505 SP=CH+32*8:FOR I=SP TO SP+7:P
OKE I,0:NEXT
510 XC=PEEK(53272):POKE 53272,(XC
 AND 240)+14
511 REM VIC VERSION: 510 POKE 52,
28:POKE 56,28:POKE 36869,255
520 SH$(0,0)="@ABC":SH$(0,1)="DEF
G":SH$(1,0)="HI":SH$(1,1)="JK"
530 SH$(2,0)="L":SH$(2,1)="M":BU$
="N":LH$="O"+CHR$(17)+CHR$(157)+"
P"
560 RE=38:RF=39:BE=23:BF=24:REM V
IC VALUES ARE 20, 21, 20, 21
565 RE(0)=RE-3:RE(1)=RE-1:RE(2)=R
E:REM USED IN MOVING DISPLAY ONLY
570 POKE 53280,11:POKE 53281,11
571 REM VIC VERSION: 570 POKE 368
79,184
590 RETURN
597 REM
598 REM SET UP VERTICAL MOVEMENT
STRINGS
599 REM
600 DIM VP$(BE):VP$(0)=CHR$(19):F
OR I=1 TO BE:VP$(I)=VP$(I-1)+CHR$
(17):NEXT
610 CL$(0)=CHR$(159):CL$(1)=CHR$(
158):CL$(2)=CHR$(152):CL$(3)=CHR$
(153)
615 CL$(4)=CHR$(155)
620 RETURN
637 REM
638 REM GENERATE RANDOM SCREEN
639 REM
640 PRINT CHR$(147):FOR I=0 TO IN
T(BE/3):N=INT(RND(9)*2):FOR J=0 T
O INT(RE/6)
645 M=INT(5*RND(9))
650 K=1+INT(RND(9)*10):ON K GOSUB
 700,670,710,670,720,670,730,670,
740,670
660 NEXT:NEXT:RETURN
670 RETURN
700 PRINT VP$(I*3)TAB(6*J)CL$(M)S
H$(0,N):RETURN
710 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*4)
715 PRINT VP$(I*3)TAB(6*J+L)CL$(M
)SH$(1,N):RETURN
720 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
725 PRINT VP$(I*3)TAB(6*J+L)CL$(M
)SH$(2,N):RETURN
730 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
735 PRINT VP$(I*3)TAB(6*J+L)CHR$(
144)BU$:RETURN
740 L=0:IF (J+1)*6<=RE THEN L=INT
(RND(9)*5)
745 PRINT VP$(I*3)TAB(6*J+L)CHR$(
5)LH$:RETURN
1000 DATA 0,0,0,3,6,255,127,31
1008 DATA 0,198,198,255,170,255,2
55,255
1016 DATA 0,48,48,254,171,255,255
,255
1024 DATA 0,0,0,0,192,248,248,240
1032 DATA 0,0,0,0,3,31,31,15
1040 DATA 0,12,12,127,213,255,255
,255
```

```
•1048 DATA 0,99,99,255,85,255,255,
 255
•1056 DATA 0,0,0,192,96,255,254,25
 2
•1064 DATA 0,0,0,0,6,15,122,63
•1072 DATA 0,0,0,0,192,224,188,252
•1080 DATA 0,0,0,0,3,7,61,63
•1088 DATA 0,0,0,0,96,240,94,252
•1096 DATA 0,0,0,0,8,28,247,127
•1104 DATA 0,0,0,0,16,56,239,254
•1112 DATA 0,0,0,0,28,28,8,62
•1120 DATA 60,66,66,60,60,60,52,60
•1128 DATA 60,44,60,60,126,90,255,
 255
```

BUG REPELLENT LINE CODES FOR RANDOM DISPLAY

LINE # 10:IL	LINE # 645:FO
LINE # 100:CK	LINE # 650:CF
LINE # 110:IG	LINE # 660:AE
LINE # 120:CF	LINE # 670:IM
LINE # 497:JD	LINE # 700:MB
LINE # 498:JD	LINE # 710:JC
LINE # 499:JD	LINE # 715:EL
LINE # 500:BD	LINE # 720:JD
LINE # 501:GN	LINE # 725:JI
LINE # 502:FM	LINE # 730:JD
LINE # 505:LP	LINE # 735:BF
LINE # 510:DN	LINE # 740:JD
LINE # 511:NB	LINE # 745:FF
LINE # 520:FB	LINE # 1000:PI
LINE # 530:EH	LINE # 1008:CL
LINE # 560:LA	LINE # 1016:NC
LINE # 565:BC	LINE # 1024:CA
LINE # 570:LK	LINE # 1032:NP
LINE # 571:NP	LINE # 1040:HA
LINE # 590:IM	LINE # 1048:HF
LINE # 597:JD	LINE # 1056:BM
LINE # 598:AH	LINE # 1064:PK
LINE # 599:JD	LINE # 1072:BE
LINE # 600:GP	LINE # 1080:OP
LINE # 610:AH	LINE # 1088:ME
LINE # 615:KH	LINE # 1096:AE
LINE # 620:IM	LINE # 1104:MG
LINE # 637:JD	LINE # 1112:OO
LINE # 638:JF	LINE # 1120:DJ
LINE # 639:JD	LINE # 1128:DO
LINE # 640:NA	LINES: 61

MOVING DISPLAY

FROM PAGE 95

```
•10 GOSUB 500:GOSUB 600
•80 GOSUB 640
•97 REM
•98 REM MAIN LOOP
•99 REM
•100 FOR K=0 TO 6:I=X(K)
•105 HP(I)=HP(I)+DI(I):IF HP(I)>RE
 (I) THEN DI(I)=-DI(I):HP(I)=RE(I)
 :GOTO 115
•110 IF HP(I)=-1 THEN HP(I)=0:DI(I
 )=-DI(I)
•115 D(I)=DI(I):IF D(I)=-1 THEN D(
 I)=0
•125 PRINT VP$(VP(I))TAB(HP(I))CL$
 (I)SH$(I,D(I))
•150 NEXT K
•160 GOTO 100
•497 REM
•498 REM SET UP CHARACTER SET--64
 VERSION
•499 REM
•500 CH=1024*14:FOR I=CH TO CH+135
 :READ N:POKE I,N:NEXT
•501 REM VIC VERSION: 500 CH=7168:
 FOR I=CH TO CH+135:READ N:POKE I,
 N:NEXT
•505 SP=CH+32*8:FOR I=SP TO SP+7:P
 OKE I,0:NEXT
•510 XC=PEEK(53272):POKE 53272,(PE
 EK(53272)AND240)+14
•511 REM VIC VERSION: 510 POKE 52,
 28:POKE 56,28:POKE 36869,255
•520 SH$(0,0)="@ABC ":SH$(0,1)=" D
 EFG":SH$(1,0)="HI ":SH$(1,1)=" JK
 "
•530 SH$(2,0)="L ":SH$(2,1)=" M":B
 U$="N":LH$="O{CD}{CL}P"
•560 RE=38:RF=39:BE=23:BF=24:REM V
 IC VALUES ARE 20,21,21,22
•565 RE(0)=RE-3:RE(1)=RE-1:RE(2)=R
 E
•570 POKE 53280,11:POKE 53281,11
•571 REM VIC VERSION: 570 POKE 368
 79,184
•590 RETURN
•597 REM
```

```
•598 REM SET UP VERTICAL MOVEMENT
 STRINGS
•599 REM
•600 DIM VP$(BE):VP$(0)="{HM}":FOR
 I=1 TO BE:VP$(I)=VP$(I-1)+"{CD}"
 :NEXT
 610 CL$(0)="{CY}":CL$(1)="{YL}":C
 L$(2)="{G2}"
•615 VP(0)=4:VP(1)=10:VP(2)=16
•620 FOR I=0 TO 2:HP(I)=1+INT(RND(
 9)*(RE(I)-2)):J=INT(RND(9)*2):IF
 J=0 THEN J=-1
•625 DI(I)=J:NEXT
•630 X(0)=0:X(1)=1:X(2)=2:X(3)=2:X
 (4)=1:X(5)=2:X(6)=2
•635 RETURN
•637 REM
•638 REM GENERATE RANDOM SCREEN
•639 REM
•640 PRINT "{SC}":FOR I=0 TO INT(B
 E/6):FOR J=0 TO INT(RE/2)
•650 K=1+INT(RND(9)*9):ON K GOSUB
•700,670,670,710,670,710,710,670,7
 10
•660 NEXT:NEXT:RETURN
•670 RETURN
•700 PRINT VP$(I*6)TAB(2*J)"{WH}"L
 H$:RETURN
•710 PRINT VP$(I*6)TAB(2*J)"{BK}{C
 D}"BU$:RETURN
•1000 DATA 0,0,0,3,6,255,127,31
•1008 DATA 0,198,198,255,170,255,2
 55,255
•1016 DATA 0,48,48,254,171,255,255
 ,255
•1024 DATA 0,0,0,0,192,248,248,240
•1032 DATA 0,0,0,0,3,31,31,15
•1040 DATA 0,12,12,127,213,255,255
 ,255
•1048 DATA 0,99,99,255,85,255,255,
 255
•1056 DATA 0,0,0,192,96,255,254,25
 2
•1064 DATA 0,0,0,0,6,15,122,63
•1072 DATA 0,0,0,0,192,224,188,252
•1080 DATA 0,0,0,0,3,7,61,63
•1088 DATA 0,0,0,0,96,240,94,252
•1096 DATA 0,0,0,0,8,28,247,127
•1104 DATA 0,0,0,0,16,56,239,254
•1112 DATA 0,0,0,0,28,28,8,62
•1120 DATA 60,66,66,60,60,60,52,60
•1128 DATA 60,44,60,60,126,90,255,
 255
```

BUG REPELLENT LINE CODES FOR MOVING DISPLAY

```
LINE # 10:IL       LINE # 615:MM
LINE # 80:CK       LINE # 620:ME
LINE # 97:JD       LINE # 625:NJ
LINE # 98:PK       LINE # 630:GH
LINE # 99:JD       LINE # 635:IM
LINE # 100:NH      LINE # 637:JD
LINE # 105:JK      LINE # 638:JF
LINE # 110:CP      LINE # 639:JD
LINE # 115:AN      LINE # 640:HH
LINE # 125:JL      LINE # 650:CJ
LINE # 150:MP      LINE # 660:AE
LINE # 160:CF      LINE # 670:IM
LINE # 497:JD      LINE # 700:LF
LINE # 498:JD      LINE # 710:GP
LINE # 499:JD      LINE # 1000:PI
LINE # 500:BD      LINE # 1008:CL
LINE # 501:GN      LINE # 1016:NC
LINE # 505:LP      LINE # 1024:CA
LINE # 510:DB      LINE # 1032:NP
LINE # 511:NB      LINE # 1040:HA
LINE # 520:BD      LINE # 1048:HF
LINE # 530:CB      LINE # 1056:BM
LINE # 560:EO      LINE # 1064:PK
LINE # 565:OE      LINE # 1072:BE
LINE # 570:LK      LINE # 1080:OP
LINE # 571:NP      LINE # 1088:ME
LINE # 590:IM      LINE # 1096:AE
LINE # 597:JD      LINE # 1104:MG
LINE # 598:AH      LINE # 1112:OO
LINE # 599:JD      LINE # 1120:DJ
LINE # 600:PL      LINE # 1128:DO
LINE # 610:BD      LINES: 63
```

CHECKLIST

FROM PAGE 20

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```
•10 GOSUB700:C0=53280:SE=54272:POK
 ESE+24,15
•20 DIMI$(10,21),G$(10,23),TI$(10,
 23),TG$(10,23),R$(24)
•25 R$(0)="{HM}":FORI=1TO24:R$(I)=
```

```
R$(I-1)+"{CD}":NEXT
•26 POKEC0+2,14:POKEC0+3,13:POKEC0
+4,5
•30 ES$=R$(24)+"
                          "
•32 A$="     ITEM     ":GOSUB1000:
A2$=B$:A3$=" C "
•33 A$=" WT ":GOSUB1000:A4$=B$
•34 ER$=R$(20)+"              "
•35 L$=CHR$(32):FORI=1TO14:C2$=L$+
C2$:NEXT:FORI=1TO4:C4$=C4$+L$:NEX
T
•36 D$=CHR$(160):C3$=D$+D$+D$:CX$=
C3$+C3$+C3$
•37 CL$=C4$+C4$+C4$
•38 A$=" WHAT ITEM? ":GOSUB1000:S2
$=B$:A$="WEIGHT OF ":GOSUB1000:S3
$=B$
•39 A$=" CODE FOR ":GOSUB1000:S4$=
B$
•40 A$=" CHOSE ANY ":GOSUB1000:S1$
=B$:A$=" DELETE ANY ":GOSUB1000:S
D$=B$
•41 A$=" F1-DEL  ":GOSUB1000:F1$=B
$:A$=" F2-CHNG ":GOSUB1000:F2$=B$
•43 A$=" F3-EXTR ":GOSUB1000:F3$=B
$:A$=" F4-TOTL ":GOSUB1000:F4$=B$
•44 A$=" F5-NEW  ":GOSUB1000:F5$=B
$:A$=" F6-COPY ":GOSUB1000:F6$=B$
•45 A$=" F7-NEXT ":GOSUB1000:F7$=B
$:A$=" F8-LIST ":GOSUB1000:F8$=B$
•46 A$="  E-END  ":GOSUB1000:E1$=B
$
•60 READ C$:IFC$="END"THEN115
•70 M=M+1:C$(M)=C$:N=0
•100 READI$,NO:IFI$="0"THENN(M)=N:
GOTO60
•110 N=N+1:I$(M,N)=I$:P=4:GOSUB110
0:G$(M,N)=A$:GOTO100
•115 PRINT"{SC}":POKE53265,PEEK(53
265)OR64
•120 POKEC0,3:POKEC0+1,3:FOR X=1TO
M:GOSUB500:PRINT"{WH}"R$(20)S1$;
•130 PRINTR$(24)F1$TAB(10)F3$TAB(2
0)F5$TAB(30)F7$;:GOSUB800
•150 GETK$:IFK$=""THEN150
•155 PRINTER$;:IFK$="{F7}"THEN260
•156 IFK$="{F1}"THENGOSUB3000:PRIN
T"{WH}"R$(20)S1$:GOTO150
•158 IFK$="{F5}"THENGOSUB3500:PRIN
T"{WH}"R$(20)S1$:GOTO150
•159 IFK$="{F3}"THENGOSUB2500:PRIN
T"{WH}"R$(20)S1$:GOTO150
•160 K=ASC(K$)-64:IFK>N(X)ORK<1THE
NGOSUB2100:GOTO150
•170 IFI$(X,K)=TI$(X,K)THENGOSUB21
00:GOTO150
•190 GOSUB400:GOTO150
•260 NEXTX:GOTO4000
•400 PRINT"{RV}{LG}"R$(1+K)TAB(13)
D$CHR$(K+96)D$;
•402 IFCHTHEN410
•405 GOSUB2000
•410 J$=I$(X,K):I$=LEFT$(J$,LEN(J$
)-1)
•420 PRINT"{RV}{GN}"TAB(17)I$;:A$=
RIGHT$(J$,1):GOSUB1000
•430 PRINT"{RV}{LG}"TAB(31)B$;TAB(
33)"{GN}"G$(X,K):IFCHTHENRETURN
•440 FT=FT+VAL(G$(X,K)):GOSUB800
•470 S(X)=S(X)+VAL(G$(X,K)):NO=S(X
):P=5:GOSUB1100:GOSUB1000
•475 PRINTR$(22)TAB(31)D$B$D$
•480 TI$(X,K)=I$(X,K):TG$(X,K)=G$(
X,K):RETURN
•500 A$=" "+CHR$(X+64):GOSUB1000:P
RINT"{SC}{WH}"TAB(13)B$D$"{BK}{RV
}"A2$"{BK}"A3$"{BK}"A4$L$
•520 Z$="":A$=C$(X):FORK=1TOLEN(A$
)+2:Z$=Z$+D$:NEXT:PRINT"{HM}{WH}{
RV}"Z$
•522 GOSUB1000:PRINT"{RV}"D$B$D$:P
RINT"{RV}"Z$
•530 PRINT"{HM}{CD}";:FORI=1TO21:P
RINTTAB(13)"{RV}"C3$C2$C3$C4$D$:N
EXT
•540 PRINT"{HM}{CD}{CD}";:FORY=1TO
N(X):A$=CHR$(Y+64):GOSUB1000:PRIN
TTAB(14)"{RV}{WH}"B$;
•545 J$=I$(X,Y):I$=LEFT$(J$,LEN(J$
)-1)
•550 PRINT"{RV}{G1}"TAB(17)I$;:A$=
RIGHT$(J$,1):GOSUB1000
•560 PRINT"{RV}{WH}"TAB(31)B$TAB(3
3)"{G1}"G$(X,Y):NEXT:RETURN
•700 POKE53280,0:POKE53281,0:PRINT
"{SC}{LG}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}";:A$=""
•715 FORI=1TO7:READA$:PRINTTAB(13)
A$:NEXT:RETURN
•750 DATA"{RV} {RO}P {RV} {RO} {RV
} {RO} {RV} {RO}P {RV} {RO}P {RV}
 {RO}N","{RV} {RO}  {RV} {RO}C{RV
} {RO} {RV} {RO}C {RV} {RO}  {RV}
{RO}M"
•760 DATA"{RV} {RO}@ {RV} {RO} {RV
```

```
} {RO} {RV} {RO}@ {RV} {RO}@ {RV}
 {RO}N",
770 DATA"    {RV} {RO}  {RV} {RO}
{RV}£{RO}MY{RV}_{RO}Y","    {RV} {
RO}  {RV} {RO} *{RV}*{RO} {RV} ",
"   {RV} {RO}@ {RV} {RO} M£ {RV}
"
780 RETURN
800 PRINTR$(5)"{BL}  TOTAL  "
810 PRINTTAB(1)C3$C3$D$:NO=FT:P=5
:GOSUB1100:GOSUB1000
820 PRINTTAB(1)"{WH}"D$B$D$:PRINT
TAB(1)C3$C3$D$:RETURN
1000 B$="":FORJ=1TOLEN(A$):B=ASC(
MID$(A$,J,1)):B=B+32:B=B-96*(B<96
)
1010 B$=B$+CHR$(B):NEXT:RETURN
1100 A$=STR$(NO):LN=LEN(A$)-1
1110 FORJ=1TOP-LN:A$=" "+A$:NEXT:
A$=RIGHT$(A$,P):RETURN
2000 POKESE+5,0:POKESE+6,247:POKE
SE+4,17:POKESE+1,50:POKESE,0
2010 FORT=1TO99:NEXT:POKESE+4,16:
RETURN
2100 POKESE+5,45:POKESE+6,165:POK
ESE+4,33:POKESE+1,6:POKESE,5
2110 FORT=1TO150:NEXT:POKESE+4,32
:POKESE+1,0:POKESE,80:RETURN
2500 I$="":N(X)=N(X)+1:POKECO,2
2502 PRINTR$(23);:INPUT"{G1} EXTR
A ITEM";I$
2504 IFI$=""THENPOKECO,3:PRINTR$(
23)"              ":RETURN
2505 IFASC(I$)=13THENPOKECO,3:RET
URN
2507 IFLEN(I$)>12THENI$=LEFT$(I$,
12)
2510 I$(X,N(X))=I$+"X":PRINTR$(23
)"                                "
2520 PRINTR$(23);:INPUT" WEIGHT";
NO:P=4:GOSUB1100:G$(X,N(X))=A$
2530 PRINTR$(23)"
      "
2540 PRINTR$(N(X)+1);:A$=CHR$(N(X
)+64):GOSUB1000:PRINTTAB(14)"{RV}
{WH}"B$;
2550 PRINT"{RV}{G1}"TAB(17)I$;:A$
="X":GOSUB1000
2590 PRINT"{RV}{WH}"TAB(31)B$;TAB
(33)"{G1}"G$(X,N(X)):GOSUB2000:PO
KECO,3:RETURN
3000 POKECO,10:PRINT"{WH}"R$(20)S
D$;
3010 GETE$:IFE$=""THEN3010
3015 E=ASC(E$)-64:IFE=-51THEN3090
3020 IFE<0ORE>N(X)THENGOSUB2100:G
OTO3010
3030 IFTI$(X,E)<>I$(X,E)THENGOSUB
2100:GOTO3010
3035 S(X)=S(X)-VAL(G$(X,E)):FT=FT
-VAL(G$(X,E)):TI$(X,E)=""
3055 PRINT"{HM}{G1}";:A$=CHR$(E+6
4):GOSUB1000:PRINTR$(E+1)TAB(14)"
{RV}{WH}"B$;
3056 J$=I$(X,E):I$=LEFT$(J$,LEN(J
$)-1)
3057 PRINT"{RV}{G1}"TAB(17)I$;:A$
=RIGHT$(J$,1):GOSUB1000
3058 PRINT"{RV}{WH}"TAB(31)B$;TAB
(33)"{G1}"G$(X,E)
3060 NO=S(X):P=4:GOSUB1100:GOSUB1
000:PRINT"{WH}"R$(22)TAB(33)B$:GO
SUB800
3090 POKECO,3:PRINT"{WH}"R$(20)S1
$:RETURN
3500 POKECO,4:PRINTR$(20)"{WH}"S2
$
3505 GETA$:IFA$=""THEN3505
3506 IFASC(A$)=13THEN3590
3510 K=ASC(A$)-64:IFK>N(X)ORK<1TH
ENGOSUB2100:GOTO3505
3520 IFI$(X,K)=TI$(X,K)THENGOSUB2
100:GOTO3505
3535 GOSUB1000:PRINT"{WH}"ER$R$(2
2)S3$B$"{G1}":INPUTW$
3540 NO=VAL(W$):P=4:GOSUB1100:G$(
X,K)=A$:PRINTR$(21)"          "
3550 PRINT"{WH}"R$(23)"      "R$(
22)S4$B$"{G1}";
3555 GETA$:IFA$=""THEN3555
3556 IFASC(A$)=13THEN3580
3560 I$(X,K)=LEFT$(I$(X,K),LEN(I$
(X,K))-1)+A$
3565 GOSUB1000:PRINT"{WH}{RV}"R$(
K+1)TAB(31)B$
3580 PRINT"{RV}{G1}"R$(K+1)TAB(33
)G$(X,K):PRINTR$(21)"       "
3590 POKECO,3:PRINT"{WH}"R$(20)S1
$R$(22)"            ":RETURN
4000 PRINT"{SC}":POKECO,3:POKECO+
1,1:POKECO+2,14:POKECO+3,15:POKEC
O+4,12
4010 A$="   CATEGORY   ":GOSUB1000
:T2$=B$:T3$="SUBTOTAL"
4015 PRINT"{RV}{G3}"R$(2)TAB(7)L$
L$L$L$T2$"{G1}"T3$;
```

```
•4020 FORI=1TOM+2:PRINT"{RV}"R$(I+
2)TAB(7)D$C3$C4$C4$C4$L$C3$C3$D$D
$:NEXT
•4030 FORI=1TOM:NO=S(I):P=5:GOSUB1
100:GOSUB1000:N2$=B$
•4040 NO=I:P=1:GOSUB1100:GOSUB1000
:N1$=B$
•4050 PRINT"{RV}{WH}"R$(3+I)TAB(8)
N1$D$TAB(12)C$(I)TAB(26)N2$:NEXT
•4055 A$=" FINAL TOTAL ":GOSUB1000
:F$=B$
•4060 NO=FT:P=6:GOSUB1100:GOSUB100
0:FT$=B$
•4090 PRINT"{WH}{RV}"R$(20)TAB(10)
CX$CX$D$D$:PRINT"{RV}"TAB(10)F$FT
$D$
•4100 PRINT"{RV}"TAB(10)CX$CX$D$D$
•4120 PRINT"{WH}"R$(24)F2$TAB(10)F
6$TAB(20)F8$TAB(30)E1$;
•4130 GETA$:IFA$=""THEN4130
•4140 IFA$="{F2}"THEN6000
•4150 IFA$="E"THENPRINT"{HM}":END
•4160 IFA$="{F8}"THEN5000
•4161 IFA$="{F6}"THEN7000
•4170 GOSUB2100:GOTO4130
•5000 PRINTES$"{HM}";:POKEC0+3,14:
POKEC0+3,13:POKEC0+4,5:POKEC0,1
•5005 TB=1:C=0:FORI=1TO23:PRINT" {
RV}"CL$D$CL$D$CL$:NEXT
•5010 PRINT"{HM}";:FORX=1TOM:A$=C$
(X):C=C+1:IFC=23THENTB=TB+13:C=1:
PRINT"{HM}";
•5011 O$="  "+C$(X):FORI=LEN(O$)TO
11:O$=O$+" ":NEXT:PRINT"{BL}"TAB(
TB)O$"{G1}"
•5015 FORY=1TON(X):IFTI$(X,Y)=""TH
EN5050
•5016 C=C+1:IFC=24THENTB=TB+13:C=1
:PRINT"{HM}";
•5020 I$=TI$(X,Y):IFLEN(I$)>12THEN
I$=LEFT$(I$,12)
•5035 PRINT"{RV}"TAB(TB)I$
•5050 NEXTY,X:PRINTR$(24)"{WH}"F2$
TAB(10)F4$TAB(20)F6$TAB(30)E1$;
•5060 GETA$:IFA$=""THEN5060
•5070 IFA$="E"THENPRINT"{HM}":END
•5080 IFA$="{F2}"THEN6000
•5090 IFA$="{F4}"THEN4000
•5095 IFA$="{F6}"THEN7000
•5099 GOSUB2100:GOTO5060
•6000 POKEC0,3:PRINT"{SC}{CD}{CD}{
CD}{CD}{CD}{G1}"TAB(12)"WHAT CATE
GORY?":PRINT
```

READER SERVICE INDEX

Page No.	Company	Reader Svc. No.
49	Academy Software	1
50	Counterpoint Software, Inc.	2
57	Creative Equipment	3
43	Micro Systems Development	4
64	Davidson & Associates	5
65	Spinnaker Software	6
65	Commodore Software	7
66	Sierra On-Line	8
66	Spinnaker Software	9
58	Spinnaker Software	10
30-35	Protecto Enterprizes	11-16
5	Cardco, Inc.	17-19
5	LETCO	20
6	MB International, Inc.	21
6	Sound Expansions	22
7	HollisCraft	23
7	TOTL Software	24
15	Atari	25
16	Broderbund	26
22	TOTL Software	27
18	Amtype Corporation	28
51	Advanced Ideas	29
54	Insta Software	30
53	Event Horizon Software	31
20	Computer Creations, Inc.	32
46	Eastern House	33
29	Microtechnic Solutions, Inc.	34
C-4	Kiwisoft	35
C-3	Cardco	36
61	Micro Ware	37
52	Cadmean Corporation	38
8	HP Books	39
24	Transactor Publishing	40
26	INDUS-TOOL	41
9	MUSE Software	42
18	Superior Micro Systems	43
10	Micro Ware	44

```
6005 FORI=1TOM:PRINTTAB(12)CHR$(I+64)" "C$(I):NEXT
6010 GETA$:IFA$=""THEN6010
6020 IFASC(A$)<65ORASC(A$)>M+64THEN6010
6026 CH=-1:POKEC0+2,14:POKEC0+3,13:POKEC0+4,5
6030 POKEC0+1,3:X=ASC(A$)-64:GOSUB500
6040 PRINT"{WH}"R$(24)F1$TAB(10)F3$TAB(20)F4$TAB(30)F5$;
6050 FORK=1TON(X):IFTI$(X,K)<>""THENGOSUB400
6051 NEXT:GOSUB800:NO=S(X):P=5:GOSUB1100:GOSUB1000:PRINTR$(22)TAB(31)D$B$D$
6060 CH=0:PRINT"{WH}"R$(20)S1$
6150 GETK$:IFK$=""THEN6150
6155 PRINTER$;:IFK$="{F4}"THEN4000
6156 IFK$="{F1}"THENGOSUB3000:GOTO6150
6157 IFK$="{F5}"THENGOSUB3500:GOTO6150
6158 IFK$="{F3}"THENGOSUB2500:GOTO6150
6160 K=ASC(K$)-64:IFK>N(X)ORK<0THENGOSUB2100:GOTO6150
6170 IFI$(X,K)=TI$(X,K)THENGOSUB2100:GOTO6150
6180 GOSUB400:GOTO6150
7000 POKEC0,1:POKEC0+1,1:PRINT"{SC}":FORI=1TO5:PRINT:NEXT:L$=CHR$(13):S$=CHR$(32)
7010 ST$=CHR$(15):FORI=1TO75:BL$=BL$+CHR$(99):NEXT
7020 INPUT"{BK}TITLE FOR HARD COPY";H$
7030 BX$=CHR$(8):FORI=1TO7:READA:BX$=BX$+CHR$(A):NEXT:BX$=BX$+ST$
7040 FORI=1TO4:READT$:T$(I)=CHR$(16)+T$:NEXT
7100 OPEN4,4:PRINT#4,CHR$(16)"25"CHR$(14)H$ST$
7105 PRINT#4,L$CHR$(18)BL$CHR$(146)L$L$
7110 FORI=1TOM:NO=S(I):P=5:GOSUB1100:GOSUB1100
7115 PRINT#4,CHR$(14)C$(I)CHR$(16)"15"A$ST$L$
7120 G=0:FORJ=1TON(I):IFTI$(I,J)=""THEN7160
7125 G=G+1:IFG=3THENG=1:PRINT#4
7130 I$=TI$(I,J):PRINT#4,T$(2*G-1)" "BX$S$LEFT$(I$,LEN(I$)-1);
7140 PRINT#4,T$(2*G)RIGHT$(I$,1)S$S$TG$(I,J);
7160 NEXT:PRINT#4,L$L$BL$:NEXT
7165 PRINT#4,L$CHR$(14)"FINAL TOTAL"CHR$(16)"15"FT;ST$
7170 PRINT#4:CLOSE4:END
9000 DATA SHELTER ,AIR MATTRESS ,490,SLPG. BAG L,1000,SLPG. BAG H,1775
9010 DATA GROUNDSHEET ,225,TENT *,1820,0,0
9020 DATA CLOTHING,TIGHTS *,275,JACKET *,260,SOCKS ,35,NYLON SHORTS ,60
9030 DATAPANTS ,340,0,0
9080 DATA ACCESSORY,LOCK+CABLE ,360,SPARE TUBE ,70,TOOLS ,205,FLASHLIGHT ,50
9085 DATASWISSKNIFE ,70,0,0
9090 DATA COOKWARE,CUP ,10,UTENSILS ,45,POT ,250,BOWL ,60
9110 DATA STOVE G,280,CARTRIDGES2,320,0,0
9200 DATAEND
9300 DATA255,193,193,193,193,193,255
9350 DATA08,24,44,60
```

BUG REPELLENT LINE CODES FOR CHECKLIST

```
LINE # 10:JH      LINE # 100:EN
LINE # 20:GN      LINE # 110:OA
LINE # 25:PA      LINE # 115:JH
LINE # 26:NH      LINE # 120:FC
LINE # 30:NL      LINE # 130:IM
LINE # 32:CM      LINE # 150:GJ
LINE # 33:KP      LINE # 155:MI
LINE # 34:BA      LINE # 156:PO
LINE # 35:LG      LINE # 158:CJ
LINE # 36:ME      LINE # 159:OL
LINE # 37:JB      LINE # 160:FL
LINE # 38:EM      LINE # 170:MI
LINE # 39:GC      LINE # 190:HM
LINE # 40:AB      LINE # 260:HA
LINE # 41:JB      LINE # 400:CG
LINE # 43:ED      LINE # 402:FF
LINE # 44:MB      LINE # 405:FL
LINE # 45:GI      LINE # 410:JE
LINE # 46:LL      LINE # 420:IJ
LINE # 60:FD      LINE # 430:EG
LINE # 70:LF      LINE # 440:IC
```

```
LINE # 470:FD
LINE # 475:FK
LINE # 480:EN
LINE # 500:DC
LINE # 520:PO
LINE # 522:DP
LINE # 530:OB
LINE # 540:MJ
LINE # 545:PK
LINE # 550:LP
LINE # 560:DE
LINE # 700:LO
LINE # 715:KC
LINE # 750:JE
LINE # 760:DP
LINE # 770:AM
LINE # 780:IM
LINE # 800:BB
LINE # 810:CP
LINE # 820:DH
LINE # 1000:EE
LINE # 1010:DC
LINE # 1100:OL
LINE # 1110:BO
LINE # 2000:EF
LINE # 2010:EH
LINE # 2100:HB
LINE # 2110:OC
LINE # 2500:CL
LINE # 2502:HI
LINE # 2504:IM
LINE # 2505:AF
LINE # 2507:AP
LINE # 2510:IN
LINE # 2520:EP
LINE # 2530:PA
LINE # 2540:IP
LINE # 2550:AA
LINE # 2590:PC
LINE # 3000:OJ
LINE # 3010:LI
LINE # 3015:BM
LINE # 3020:PN
LINE # 3030:NL
LINE # 3035:MP
LINE # 3055:PB
LINE # 3056:EO
LINE # 3057:LP
LINE # 3058:BL
LINE # 3060:EL
LINE # 3090:IP
LINE # 3500:BG
LINE # 3505:LB
LINE # 3506:KP
LINE # 3510:DK
LINE # 3520:MG
LINE # 3535:MD
LINE # 3540:KJ
LINE # 3550:HB
LINE # 3555:LE
LINE # 3556:JC
LINE # 3560:GB
LINE # 3565:BO
LINE # 3580:OH
LINE # 3590:IA
LINE # 4000:KO
LINE # 4010:GN
LINE # 4015:DN
LINE # 4020:KB
LINE # 4030:KC
LINE # 4040:BG
LINE # 4050:CE
LINE # 4055:NM
LINE # 4060:ED
LINE # 4090:DD
LINE # 4100:HF
LINE # 4120:OE
LINE # 4130:LM
LINE # 4140:PF
LINE # 4150:JP
LINE # 4160:PF
LINE # 4161:NC
LINE # 4170:OA
LINE # 5000:AO
LINE # 5005:NI
LINE # 5010:HK
LINE # 5011:ML
LINE # 5015:HO
LINE # 5016:NK
LINE # 5020:CL
LINE # 5035:LL
LINE # 5050:JL
LINE # 5060:KF
LINE # 5070:JP
LINE # 5080:PF
LINE # 5090:PM
LINE # 5095:NC
LINE # 5099:NB
LINE # 6000:BM
LINE # 6005:KN
LINE # 6010:KD
LINE # 6020:ID
LINE # 6026:IL
LINE # 6030:DO
LINE # 6040:DP
LINE # 6050:OB
LINE # 6051:EJ
LINE # 6060:GJ
LINE # 6150:LE
LINE # 6155:OE
LINE # 6156:NI
LINE # 6157:EB
LINE # 6158:OD
LINE # 6160:NA
LINE # 6170:LF
LINE # 6180:KC
LINE # 7000:HL
LINE # 7010:GI
LINE # 7020:OG
LINE # 7030:IA
LINE # 7040:OE
LINE # 7100:EP
LINE # 7105:BB
LINE # 7110:HD
LINE # 7115:KH
LINE # 7120:KH
LINE # 7125:BB
LINE # 7130:OO
LINE # 7140:AG
LINE # 7160:AC
LINE # 7165:EL
LINE # 7170:KH
LINE # 9000:IH
LINE # 9010:FP
LINE # 9020:EG
LINE # 9030:HI
LINE # 9080:CK
LINE # 9085:AK
LINE # 9090:JC
LINE # 9110:LH
LINE # 9200:GB
LINE # 9300:PN
LINE # 9350:LC
LINES: 185
```

RENUMBERING

FROM PAGE 29

IMPORTANT

Before typing in an *Ahoy!* program, refer to the first two pages of the program listings section.

```
•100 REM 2-17-84
•110 DIM L(2000)
•120 PRINT"{SC}{CD}";SPC(9);"RENUM
BER UTILITY":PRINT
•130 PRINTSPC(8)"(C) COPYRIGHT 198
4":PRINTSPC(11);"ANTHONY WOOD"
•140 PRINT:INPUT"NAME OF PROGRAM T
O RENUMBER";I$
•150 INPUT"NAME OF RENUMBERED PROG
RAM";P$
•160 INPUT"FIRST LINE NUMBER? 100{
CL}{CL}{CL}{CL}{CL}";S
•170 INPUT"INCREMENT? 10{CL}{CL}{C
L}{CL}";I:PRINT
•180 DEF FNMB (X)=INT(X/256)
•190 DEF FNLB (X)=X-FNMB(X)*256
•200 PRINT"PASS ONE...":OPEN15,8,1
5,"I0"
•210 OPEN2,8,2,I$+",P,R"
•220 INPUT#15,E,ER$,A,A:IFE<>0THEN
PRINTER$:CLOSE15:END
•230 GOSUB 490:A=V
•240 GOSUB 490:IFV=0THEN 280
•250 E=V:GOSUB 490:L(K)=V:K=K+1:A=
```

```
A+4
260 IFA=ETHEN 240
270 GET#2,A$:A=A+1:GOTO 260
280 CLOSE 2:PRINT"PASS TWO..."
290 OPEN2,8,2,I$+",P,R":OPEN3,8,3
,P$+",P,W":INPUT#15,E,ER$,A,A
300 IFE<>0THENPRINTER$:CLOSE2:CLO
SE15:END
310 GOSUB 490:A=V:PRINT#3,A$;B$;:
C=0
320 REM READ LINE & CHANGE REF'S
330 GOSUB 490:GOSUB 490:L$=""
340 GET#2,A$
345 IFA$=CHR$(137)THENGOSUB500:GO
TO 345
350 IFA$=CHR$(141)THENGOSUB500:GO
TO 345
360 IFA$=CHR$(167)THENGOSUB500:GO
TO345
370 IFLEN(A$)=0THEN 430
380 IFA$<>CHR$(24)THENL$=L$+A$:GO
TO 340
390 L$=L$+A$
400 GET#2,A$:IFLEN(A$)=0THEN 430
410 L$=L$+A$:IFA$=CHR$(24)THEN 34
0
420 GOTO 400
430 A=A+5+LEN(L$):PRINT#3,CHR$(FN
LB(A));CHR$(FNMB(A));
440 PRINT#3,CHR$(FNLB(S+C*I));CHR
$(FNMB(S+C*I));
450 PRINT#3,L$;CHR$(0);
460 C=C+1:IFC<>KTHEN 330
470 PRINT#3,CHR$(0);CHR$(0);CHR$(
0);
480 CLOSE2:CLOSE3:CLOSE15:PRINT"D
ONE":END
490 GET#2,A$:GET#2,B$:V=ASC(A$+CH
R$(0))+ASC(B$+CHR$(0))*256:RETURN
500 N$="":L$=L$+A$
510 GET#2,A$:IFA$=" "THEN 510
520 GOTO 540
530 GET#2,A$
540 IFA$<"0"ORA$>"9"THEN 560
550 N$=N$+A$:GOTO 530
560 IFN$=""THEN610
570 N=VAL(N$):FORX=0TOK-1
580 IFL(X)=NTHEN 600
590 NEXT:PRINT"UNDEF'D LINE NO: "
;N$:L$=L$+"0000":GOTO610
600 L$=L$+STR$(X*I+S)
610 IFA$=","THEN500
630 RETURN
```

BUG REPELLENT LINE CODES FOR RENUMBERING UTILITY

LINE # 100:PK	LINE # 370:GL
LINE # 110:OE	LINE # 380:IN
LINE # 120:DA	LINE # 390:LK
LINE # 130:NH	LINE # 400:CO
LINE # 140:PI	LINE # 410:EN
LINE # 150:OF	LINE # 420:CA
LINE # 160:EL	LINE # 430:GK
LINE # 170:NN	LINE # 440:GJ
LINE # 180:MI	LINE # 450:AK
LINE # 190:MK	LINE # 460:OI
LINE # 200:MG	LINE # 470:IA
LINE # 210:BN	LINE # 480:AJ
LINE # 220:EH	LINE # 490:CC
LINE # 230:ME	LINE # 500:GL
LINE # 240:IA	LINE # 510:CP
LINE # 250:PJ	LINE # 520:CN
LINE # 260:AP	LINE # 530:JB
LINE # 270:GF	LINE # 540:KO
LINE # 280:NP	LINE # 550:BM
LINE # 290:OH	LINE # 560:EH
LINE # 300:IM	LINE # 570:EF
LINE # 310:OI	LINE # 580:OM
LINE # 320:AK	LINE # 590:DO
LINE # 330:CF	LINE # 600:KP
LINE # 340:JB	LINE # 610:GP
LINE # 345:DO	LINE # 630:IM
LINE # 350:MP	LINES: 54
LINE # 360:EH	

CREATING YOUR OWN GAMES

Continued from page 17

marker, contained in the array RE(*x*). This is set up in line 565.

The setup in lines 600 to 635 is more complex now, too. Each ship has its own color (line 610), and each ship has its own row number (line 615,

the array VP*(x)*). Line 620 randomly assigns each ship its starting position along its row—HP*(x)*—and its starting direction of movement—DI*(x)*. Thus, the ocean liner will be SH$(0,*x*), its horizontal position will be HP(0), its vertical position will be VP(0), its color will be CL$(I), and its direction of movement will be DI(0). This means that we can use the same statement to PRINT any of the ships, just by changing a single index variable, *I:*

```
125 PRINT VP$(VP(I))TAB(HP(I))CL$
(I)SH$(I,D(I))
```

The same lines can also be used to change the horizontal position and direction as each ship moves back and forth on its row:

```
105 HP(I)=HP(I)+DI(I):IF HP(I)>RE
(I) THEN DI(I)=-DI(I):HP(I)=RE(I)
:GOTO 115
110 IF HP(I)=-1 THEN HP(I)=0:DI(I
)=-DI(I)
115 D(I)=DI(I):IF D(I)=-1 THEN D(
I)=0
```

Notice that there are *two* variables to control direction, DI*(x)* and D*(x)*. This is because direction is used two ways. DI(I) is used to change the value of HP(I). Therefore, to move leftward, DI(I) must have a value of −1, and a value of 1 to move rightward. However, in PRINTing SH$(I,D(I)), the second element of the array must be a positive integer—a value of −1, for leftward movement, would cause an error. Therefore, when the variable DI(I) is set to −1, the variable D(I) is set to 0. A little complicated, but it works.

Only one more complication. We could move all three ships at the same speed by having the main loop be

```
100 FOR I=0 TO 2
```

Then each time through the loop, one of the ships would move. But we want each movement of the ocean liner to be accompanied by two movements of the cruiser and four movements of the speedboat. Therefore we use the line

```
100 FOR K=0 TO 6:I=X(K)
```

The array X*(x)* has been previously set up (in line 630) so that X(0) equals 0, X(1) and X(4) equals 1, and all the other values equal 2. Thus, each time the program completes the *K* loop, it has moved the ocean liner once (when K equaled 0), the cruiser twice (when K equaled 1 and when K equaled 4), and the speedboat four times (when K equaled 2, 3, 5, and 6). There is a little jerkiness in the movement, but when you run the program you'll see that BASIC can get pretty quick movement.

Actually, there's a trick to this. By having the speedboat move so often, the screen is given the illusion of speed—even though the other two ships move much, much more slowly. Remember, the illusion of speed is all that matters.

YOU CAN DO BETTER THAN THIS

This is really a very simple, unsophisticated display. Three types of boats, a buoy, and a lighthouse aren't exactly a dazzling effect. But you should see that movement on the screen and displays with randomness and variety are easy to program.

And for those of you who have fun with this display routine, how about writing the few lines of code that would get the player's boat on the screen, drifting steadily upward, while the player controls the boat's left-and-right movement? Send in your alterations to either the *Moving Display* or *Random Display* programs, either on disk or carefully written out, and we'll run the best implementation of the game in a later issue.

SEE PROGRAM LISTINGS ON PAGE 86

THE FILE SLEUTH

Continued from page 60

used instead of the "1". The computer would then automatically write an "end-of-file" character to the end of our file when it is closed, but for this example it doesn't really matter.

The secondary address for the disk can be any number from 2 to 14 when we are reading or writing a disk data file. Since the choice is arbitrary, an "8" was used for consistency.

The string following the secondary address contains the file-name and (for the disk) other file information. The file-name is used only when we want the computer to locate the file. Once it has been located, we always refer to the file by means of its file-number. For disk files, we included an "S" and a "W" to indicate that this file is a "Sequential" file and that we are "Writing" to it.

The CLOSE statement is much simpler to explain. The number following it tells which file we are done working with. Once again, we are referring to the file by means of its file-number. When the file was OPENed, the computer set aside a buffer in memory through which the data we wrote was gathered and transferred to the file. The CLOSE statement makes sure that nothing is left behind in the buffer and that the proper "end-of-file" information is sent to the tape or disk. It may be impossible to get any or all of the data back out of a file which was not properly CLOSEd.

It is good programming practice to leave files open only as long as necessary to access them. Although the computer doesn't flag a missing CLOSE statement as it does a missing NEXT, for example, it is a good idea to make sure that every OPEN statement has a corresponding CLOSE statement which will always be executed. If a file-handling program you are working on "bombs" and a file is left open, the red light on the disk drive may remain on. If so, it is advisable to close any files that might have been opened. You can do this in the direct mode simply by typing CLOSE 8 for example. Tape files are not as vulnerable to mishandling as disk files, but you may as well use proper techniques in all your programming.

ONWARD AND ONWARD

Whew! Who would have thought that a three-line program would have required so much explanation? Certainly there is a lot to learn about the use of files, but we have already covered the majority of the new concepts. If you understand what we've done so far, then you are ready to continue. If not, go through the previous section one more time.

Let's find out how to bring the data from the file back into the computer. There are two basic ways of doing that. We may GET the data from the file one character at a time, or we may INPUT a complete data record.

We'll begin with the shorter method. This program INPUTs the data in one chunk and assigns it to the variable D$:

```
40 OPEN 1,1,0,"FIRSTFILE"
50 INPUT#1, D$
60 PRINT D$
70 CLOSE 1
```

If you are using the disk, change line 40 to:

```
40 OPEN 8,8,8,"FIRSTFILE,S,R"
```

and change the 1's in lines 50 and 70 to 8's.

Hopefully the OPEN and CLOSE statements are reasonably familiar. We now use a 0 for the secondary address to specify a "read" operation on the tape file. The numbers in the OPEN statement for the disk don't change, but the "R" for "read" replaces the "W" which we used previously to write the file.

Here are just a couple of pointers before we continue. Although it is generally preferable for readability to separate terms within a program, there are some cases where it is not possible. Specifically the PRINT# and INPUT# (as well as the GET#) statements must not have a space before the "#". Another thing to remember is that the number used with the INPUT#, PRINT#, and GET# as well as the CLOSE statements is always the first number in the OPEN statement. That makes it easy.

Before you run this program, rewind the tape to the start of the file (zero on the counter if you remembered to reset it). Also either erase lines 10-30 from the first program, or add the statement:

```
5 GOTO 40
```

When you type RUN with the tape version, the computer responds "Press Play on Tape." If you do just that, the recorder runs for a few seconds, accompanied by blankings of the video screen. Finally the word "COMPUTER" is displayed on the screen followed by the READY prompt. The disk version gives the same results without the blanking of the screen.

REALLY GETTING INTO FILES

Now let's look at a slightly more complicated but potentially more powerful method of inputting data from a file. The GET# statement, just like its keyboard counterpart GET, reads data one character at a time. This program gives us some insight as to how a file is organized:

```
40 OPEN 1,1,0,"FIRSTFILE"
50 GET#1, C$
60 PRINT C$,ASC(C$)
70 IF C$ < > CHR$(13) THEN 50
80 CLOSE 1
```

If you are using a disk, change line 40 (as before) to:

```
40 OPEN8,8,8,"FIRSTFILE,S,R"
```

Also change the 1's in lines 50 and 80 to 8's.

Properly position the tape and type RUN. Vertically down the screen are the characters of the word "COMPUTER" and beside each letter is its ASCII code. The last ASCII code is 13, which represents a carriage return. Just as a normal PRINT statement puts a carriage return at the end of a statement like

```
PRINT "COMPUTER"
```

the PRINT# statement which created this file also put a CHR$(13) at the end of the file.

This type of program lets you examine the details of any sequential file. Each record in a file is separated from the next by means of some terminator or separator. This separator will be either a comma, CHR$(44), a carriage return, CHR$(13), or a semicolon, CHR$(59).

If you know the form of the PRINT# statement(s) which created the file, you may use INPUT# statement(s) of exactly the same format. If, however, you are working with a file, and you don't know how the records are grouped, the GET# statement can help you dig out that information.

Line 60 in the program above has a possible glitch. If a character with ASCII code 0 is read into C$ from the file, ASC(C$) will generate an ?ILLEGAL QUANTITY ERROR, since character 0 is read as a null string instead of a true CHR$(0). (Try PRINT ASC("") to see what happens.) To protect against such a case, use ASC(C$+CHR$(0)) as explained on page 342 of the *Commodore 64 Programmer's Reference Guide*.

You might add lines 75 and 100 to the program above:

```
75 R$ = R$ + C$
100 PRINT R$
```

This way you may build up the string character by character and display the results.

The following program is a more general sequential file investigator:

```
5 REM >> GENERAL FILE READER <<<
10 INPUT"WHAT FILE NAME";N$
20 OPEN 8,8,8,N$+",S,R"
30 GET#8,A$
40 A=ASC(A$+CHR$(0))
50 PRINT A$,A
60 C$=C$+A$
70 REM - CHECK FOR CARRIAGE RETUR
N OR COMMA OR SEMI-COLON
80 IF A<>13 AND A<>44 AND A<>59 T
HEN 30
90 PRINT "---STRING VALUE = ";C$
100 N=VAL(C$)
110 PRINT "--NUMERIC VALUE N = ";
N
120 PRINT "              CHR$(N) =
  ";CHR$(N)
130 C$=""
140 PRINT:PRINT"PRESS F1 TO STOP/
/ANY OTHER KEY TO GO"
150 GET KB$:IF KB$="" THEN 150
160 IF KB$<>CHR$(133) THEN 30
170 CLOSE 8:END
```

This program reads each character of a specified sequential file. It builds up a string from the characters until a carriage return, a comma, or a semicolon is found. It prints the string, the numeric value of the string, and the ASCII character with that numeric value.

If you hold down the space bar, the entire file is displayed. You may stop the program and close the file by pressing f1. If the program reads beyond the end of the file, the screen scrolls continuously since no more separators are found. Press the RUN/STOP key to stop the program if that happens.

For tape files, change line 20 as follows:

```
20 OPEN 1,1,0,N$
```

Also change the 8's in lines 30 and 170 to 1's. With tapes it is not necessary to specify the filename. If you just press <RETURN>, the program will read the first file it finds (either data or program files). If you have both disks and tapes, be sure that the disk Wedge is not installed when you work with tape files. There are incompatibilities between the two. If the Wedge is installed, type @Q to exit it.

For more advanced investigations into files, you might change the "S" in line 20 to "P" in the disk version. This will allow you to read and display the contents of program files rather than data files. The tape version will not need any changes to read either type of file. You might want to put a statement similar to line 150 after line 30 so that you can more easily step through the file. You will find that the separators that line 80 looks for are less common in program files. Also you may want to "filter" out some of the character codes which will disrupt the color setup on your display. (A color combination of white letters on a white background would provide less than useful information.)

Hopefully you have learned enough about files to be able to write and then read the confidential file described at the beginning of this article. Unfortunately, that will get you only part of the way to the goal. There is still the problem of deciphering the file once you have read it.

Put your cryptoanalytical powers and file-handling abilities to work. See if you can write a program which will read Confidential File 3X501 and properly decode it. You will know when your answer is correct! □

EDUCATIONAL SOFTWARE

Continued from page 66

Inabeth Miller, the librarian for the faculty of education at Harvard, says that in the schools, Logo will be one of the two most important pieces of educational software, along with word processing programs. Commodore has its own version of Logo for the C-64. It includes turtle graphics commands and also has sophisticated programming features for manipulating text. Another turtle graphics environment is available in Tamarack Software's *Vanilla PILOT.* Short for programmed inquiry, learning or teaching, PILOT began at the University of California Medical Center as an authoring language for computer-assisted instruction. When Logo became popular, versions of PILOT with turtle graphics were developed. Today, the line between the two languages has faded.

Spinnaker's president, David Seuss, praises turtle graphics but finds little value in the text manipulation capabilities of Logo. He feels that those capabilities, founded on research in artificial intelligence, are unnecessary overhead for Logo's turtle graphics. These opinions are evident in Spinnaker's *Delta Drawing:* a turtle graphics environment for even very young children. Single keystrokes—D for draw, R for turn right—make the turtle respond immediately. Your child can program by watching the turtle rather than by typing a list of commands which 'should' do what he wants. In *Delta Drawing,* he can toggle to a screen which lists all the commands he has typed, but children usually start by just watching the turtle obey their commands.

Whatever category of educational software you choose for your child— drill and practice, tutorial, simulation, or problem solving—consider the child's interests, needs and abilities. If all three areas are weak (the child has no interest in the subject, doesn't need to know it, and lacks the ability to learn it), move on. It is as often the presentation of the subject which peaks a child's interest as it is the subject. "This program looks neat!" If a particular method of presentation can encourage your child to learn, more power to it.

Programs mentioned in this article:

Chaterbee, Tronix, 8295 South La Cieniga Boulevard, Culver City, CA 90301 (213-215-0529). $39.95; disk.

Commdore Logo (grades 2 and up), Commodore, Inc., 1200 Wilson Drive, West Chester, PA 19380 (215-431-9100). $59.00; disk.

Computer Preparation/SAT, Harcourt Brace Jovanovich, Inc., 757 Third Avenue, New York, NY 10017 (212-888-4303). $79.95 for 2 disks and workbook.

Delta Drawing and *Trains,* Spinnaker Software Corp., 215 First Street, Cambridge, MA 02142 (617-868-4700). $39.95; cartridge.

Dragon's Keep (grades 2 to 4), Sierra On-Line, Inc., Sierra On-Line Building, Coarsegold, CA 93614 (219-683-6858). $29.95; disk.

Early Music Games, Counterpoint Software, Inc., Suite 140 Shelard Plaza North, Minneapolis, MN 55426 (612-544-4720). $29.95; disk.

The Factory (grades 4 to adult), Sunburst Communications, Inc., 39 Washington Avenue, Pleasantville, NY 10570 (800-431-6616). $49.00 disk with backup and teacher's guide.

Fay: The Word Hunter (grades 3 to 10), Didatech Software, 549-810 West Broadway, Vancouver, B.C., Canada V5Z 4C9 (604-687-3468). $34.95; disk.

Logo Locomotion (requires *Commodore Logo*), Computer Literacy Associates, P.O. Box 254, Sherborn, MA 01770 (617-655-4856). $23.95; disk.

SAT Review, Moses Engineering, P.O. Box 1 1038, Ardmore Highway, Huntsville, AL 35805 (205-837-3356). $30.00 for 5 cassettes.

Space Rescue, HesWare, 150 North Hill Drive, Brisbane, CA 94005 (800-624-2442). $39.95; disk.

Special Learning Ed Software (spelling) (grades 4 and up), SLED Software, P.O. Box 16322, Minneapolis, MN 55416 (612-926-5820). $199 for 2 disks or 26 tapes.

Speed Reader II (all grades), Davidson & Associates, 6069 Groveoak Place # 12, Rancho Palos Verdes, CA 90274 (213-373-9473). $69.95 for program disk and 1 data disk, $19.95 for additional data disks.

Vanilla PILOT, Tamarak Software, Darby, MT 59829 (406-821-4596) $29.95; cassette or disk.

CARDCO Letter Quality Printers

"Commodore-ready", PLUS . . .

An excellent line of Letter Quality PRINTERS are ready for you and "Commodore-ready", PLUS . . . compatible input for PC, PC jr., TRS-80 and a host of other personal computers.

All three CARDCO PRINTERS are true letter quality printers, bi-directional with continuous printing capability, for computers with centronics parallel output, of excellent quality, standard friction-feed, designed in attractive neutral to blend with any decor in your office, home or business. The fine letter quality is suited for personal or business use, ideal for word processing applications including subscript, supersubscript and bold-typing.

Priced attractively and much lower than you expect.

- **Model LQ/1 . . .** 13" carriage, daisy wheel type, printing speed 14 cps.
- **Model LQ/2 . . .** 8½" carriage, drumhead design, printing speed 12 cps.
- **Model LQ/3 . . .** 11" carriage, daisy wheel type, printing speed 13 cps.

Write for illustrated literature and prices or see CARDCO Computer Accessories and Software wherever Computers are sold.



cardco, inc.
300 S. Topeka Wichita, Kansas 67202 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

Reader Service No. 36